

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"
УДК _____

«До захисту допущено»
Завідувач кафедри
_____ О.В. Коваль
(підпис) (ініціали, прізвище)
“ ___ ” _____ 2018р.

Магістерська дисертація

зі спеціальності - 122 Комп'ютерні науки та інформаційні технології
за спеціалізацією - Геометричне моделювання в інформаційних системах
на тему: Система корегування навчального плану вищих навчальних закладів в залежності від потреб ринку інформаційних технологій

Виконав (-ла): студент (-ка) 6 курсу, групи ТР-71М
Хохлова Яніна Геннадіївна
(прізвище, ім'я, по батькові)

_____ (підпис)

Науковий керівник к.т.н., доцент Гагарін.О.О.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Рецензент _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ - 2018

**Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 122 Комп'ютерні науки та інформаційні технології
за спеціалізацією - Геометричне моделювання в інформаційних системах

ЗАТВЕРДЖУЮ
Завідувач кафедри
Коваль О.В. _____
(прізвище, ініціали) (підпис)
« ____ » _____ 2018р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

Хохлова Яніна Геннадіївна
(прізвище, ім'я, по батькові)

1. Тема дисертації Система корегування навчального плану вищих навчальних закладів в залежності від потреб ринку інформаційних технологій

Науковий керівник к.т.н., доцент Гагарін О.О.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “ ” _____ 201 року №

2. Строк подання студентом дисертації “ ” _____ 201 року

3. Об'єкт дослідження невідповідність навчальних планів вищих навчальних закладів до потреб ринку інформаційних технологій

4. Предмет дослідження комп'ютерні інформаційні технології корегування навчальних планів в залежності від потреб ринку інформаційних технологій.

5. Перелік питань, які потрібно розробити _____

1) проаналізувати сучасні системи найму персоналу;

2) проаналізувати методи порівняння навиків випускника та потреб ринку;

3) розробити метод оцінки важливості навиків;

4) розробити архітектуру веб-застосунку;

5) розробити програмне забезпечення.

6. Орієнтований перелік ілюстративного матеріалу _____

мета, постановка задачі, алгоритм порівняння навиків випускника та потреб ринку, алгоритм визначення важливості навиків, організація структури програмного забезпечення

7. Орієнтований перелік публікацій _____
 «Засоби корегування планів навчання до потреб ІТ ринку праці» // Сучасні проблеми наукового забезпечення енергетики: Матеріали XVI Міжнародної науково-практичної конференції аспірантів, магістрантів і студентів, м. Київ, 24–27 квітня 2018 р. У 2 т. – К. : 7 КПІ ім. Ігоря Сікорського», 2018. – Т. 2. – 147с.

8. Дата видачі завдання « ».

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Отримання завдання		
2	Збір інформації		
3	Аналіз вимог завдання, вибір методів і засобів розв'язання поставленої задачі		
4	Підготовка матеріалів магістерської роботи		
5	Проміжний контроль підготовки		
6	Підготовка публікацій		
7	Підготовка доповідей на конференціях за темою магістерської роботи		
8	Доповідь на конференції		
9	Написання основних розділів автореферату		
10	Звіт за перший рік роботи над магістерською дисертацією		

Студент

(підпис)

Хохлова Я.Г.

(прізвище та ініціали)

Науковий керівник

(підпис)

Гагарін О.О.

(прізвище та ініціали)

РЕФЕРАТ

Структура й обсяг дипломної роботи. Магістерська дисертація складається зі вступу, п'ятьох розділів, висновку, переліку посилань з 51 найменування, 2 додатки, і містить 23 рисунки, 23 таблиць. Повний обсяг магістерської дисертації складає 96 сторінок, з яких перелік посилань займає 4 сторінки, додатки – 6 сторінок.

Актуальність теми. Ринок інформаційних технологій потребує велику кількість спеціалістів. Інформаційні технології пронизують всі сфери життя. Через це виникає необхідність спеціалістів різних напрямків зі знанням специфічних технологій. На жаль, роботодавці стверджують, що рівень знань випускників вузу занадто низький, для того щоб зразу піти працювати. Розробка системи корегування навчального плану вищих навчальних закладів в залежності від потреб ринку інформаційних технологій допоможе покращити якість знань випускників вузів.

Мета дослідження полягає в створенні засобів та виявленні нових підходів корегуванню навчальних планів в залежності від потреб ринку інформаційних технологій.

Об'єктом дослідження є невідповідність навчальних планів вищих навчальних закладів до потреб ринку інформаційних технологій

Предметом дослідження є комп'ютерні інформаційні технології корегування навчальних планів в залежності від потреб ринку інформаційних технологій

Наукова новизна одержаних результатів. Найбільш суттєвими науковими результатами магістерської дисертації є:

- алгоритм порівняння навиків випускника та потреб ринку інформаційних технологій
- метод оцінки важливості навиків для отримання першої роботи

Практичне значення. Приведений алгоритм порівняння навиків випускника та потреб ринку інформаційних технологій дозволяє розробити комплекс програмних засобів, що забезпечує достатньо повне і цілісне рішення задач

корегування навчальних планів в залежності від потреб ринку інформаційних технологій.

Апробація результатів. Основні положення роботи доповідались та обговорювались на:

1. XV Міжнародній науково-практичній конференції аспірантів, магістрів, студентів «Сучасні проблеми наукового забезпечення енергетики» (Київ, 24-27 квітня 2018р.);

Публікації. Наукові положення дипломної роботи опубліковані у одній роботі

Основні публікації по темі дисертації

1. «Засоби корегування планів навчання до потреб ІТ ринку праці» // Сучасні проблеми наукового забезпечення енергетики: Матеріали XVI Міжнародної науково-практичної конференції аспірантів, магістрантів і студентів, м. Київ, 24–27 квітня 2018 р. У 2 т. – К. : 7 КПІ ім. Ігоря Сікорського», 2018. – Т. 2. – 147с.

Ключові слова: *ЯКІСТЬ ОСВІТИ, МОДЕЛЬ СПЕЦІАЛІСТА, ВИМОГИ РОБОТОДАВЦЯ, ІНФОРМАЦІЙНІ СИСТЕМИ*

ЗМІСТ

Перелік умовних скорочень і позначень	9
Вступ	11
1. Обґрунтування потреби створення системи корегування навчального плану в залежності від потреб ринку інформаційних технологій.....	14
1.1. Огляд існуючих систем моніторингу ринку ІТ.....	15
1.2. Відні та вихідні дані	21
1.3. Висновки до розділу 1 та постановка задачі.....	21
2. Проектування системи корегування навчального плану в залежності від потреб ринку інформаційних технологій	23
2.1. Функціональні можливості системи.....	26
2.2. Сховища даних	27
2.3. Алгоритм оцінювання необхідності навиків	29
2.4. Алгоритм порівняння потреб ринку та існуючих навиків.....	30
2.5. Регулярні вирази.....	32
2.6. Висновки до розділу 2	34
3. Опис програмної реалізації	35
3.1. Засоби розробки.....	35
3.1.1. Основна архітектура проекту.Паттерн MVC	35
3.1.2. Фреймвор. NET CORE	40
3.1.3. Мова розмітки HTML.....	42
3.1.4. Мова програмування JavaScript	44
3.1.5. Формат передачі даних JSON.....	44
3.1.6. Технологія комунікації сервера та клієнта AJAX	46
3.1.7. Технологія Razor	48
3.1.8. Середовище розробки MS Visual Studio 2017	50
3.1.9. Система управління базами даних Microsoft SQL Server Express.....	51

3.2. Опис бази даних.....	52
3.3. Висновки до розділу 3	55
4. Взаємодія з користувачем	56
4.1 Інсталяція та налаштування програмного продукту.....	57
4.2 Опис програмного продукту.....	57
4.3 Сценарії роботи користувача з системою.....	58
4.4. Висновки до розділу 4	65
5. Стартап проект.....	66
5.1 Опис ідеї проекту	66
5.2 Технологічний аудит ідеї проекту	69
5.3 Аналіз ринкових можливостей запуску стартап-проекту.....	70
5.4 Розроблення ринкової стратегії проекту	79
5.5 Аналіз ринкових можливостей запуску стартап-проекту.....	81
5.6. Висновки до розділу 5	83
Висновки.....	84
Список використаних джерел.....	86
Додаток А.....	91
Додаток Б Акт впровадження	95

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ПОЗНАЧЕНЬ

СД – сховище даних

БД – база даних

ETL – Extract Transform Load, підхід до обробки даних

СКДБ – Система керування базами даних

OLTP – Online Transaction Processing, онлайн нова обробка транзакцій

Фреймворк - інфраструктура програмних рішень, що полегшує розробку складних систем

Back End – розробка серверної частини та взаємодії з базою даних

Front End – це інтерфейс для взаємодії між користувачем і back end.

ВСТУП

Ринок інформаційних технологій потребує велику кількість спеціалістів. Інформаційні технології пронизують всі сфери життя. Через це виникає необхідність спеціалістів різних напрямків зі знанням специфічних технологій. Розробка системи корегування навчального плану вищих навчальних закладів в залежності від потреб ринку інформаційних технологій допоможе покращити якість знань випускників вузів.

Сучасний ринок праці почав висувати абсолютно нові вимоги до високкваліфікованих спеціалістів [1]. В пріоритеті стає якість підготовки майбутнього професіонала. Проблема якості підготовки високкваліфікованих спеціалістів і факторів, які зумовлюють цей процес, завжди викликало розбіжності між наявними спеціалістами, замовниками та виконувачами.

На один щабель з якістю навчання необхідно ставити визначення критеріїв і індикаторів процесу оцінювання [2]. У визначенні критеріїв якості вищої професійної освіти можна виділити такі фактори як, затребуваність спеціаліста, працевлаштування після закінчення вузу.

Працевлаштування випускника є важливою соціально - економічною проблемою, як для освітнього процесу в Україні, так і всієї країни в цілому. Щоб вирішити цю проблему, необхідно створити механізми взаємодії роботодавців та вищих навчальних закладів. В процесі підготовки спеціаліста необхідно враховувати перспективи їх подальшого працевлаштування на основі реальних потреб ринку інформаційних технологій.

Вимоги до спеціалістів в області інформаційних технологій маєть тенденції до швидкого росту до розвитку. Це пов'язано з потужним впровадженням інформаційних технологій в бізнес, фінанси, виробництві і в повсякденному житті. Об'єм знань, який необхідний для працевлаштування в сфері інформаційних технологій буде постійно рости і змінюватися. Необхідно постійно відслідковувати

потреби ринку інформаційних технологій та знаннями, які надає вищий навчальний заклад студентам [3].

Існує проблема працевлаштування випускників вищих навчальних закладів. Це пов'язано з тим, що є відсутність практичних навичок. У 90% вакансій можна побачити вимогу, що необхідно мінімум 1 рік практичного досвіду роботи. Роботодавці кажуть, що випускники вузів не мають необхідної підготовки.[4] Молоді спеціалісти вимагають для себе багато уваги і часу для навчання необхідним навикам. Найм на роботу такої людини схожий на видачу кредиту довіри. Ніхто не дає гарантій, що новий співробітник зможе виправдати вкладений в нього час та увагу.

Сфера інформаційних технологій відрізняється надшвидкими темпами розвитку. Через таку особливість знання, які отримують студенти можуть застарівати ще до закінчення ними навчального процесу [5]. Ця сфера найбільш за всі інші потребує від спеціаліста саме практичних навичок та досвід роботи над реальними проектами.

Для знаходження талантів, які будуть підходити саме певній компанії і відсіяти всіх інших роботодавці, змушені проводити IT-курси / інтернатури / програми. Вони інвестують умовну тисячу доларів в людину, щоб з часом заробити на ній набагато більше.[6] Але вони могли вкладати ці кошти в вузи і отримувати гарних спеціалістів, а вузи, в свою чергу, змогли б розвиватися і покращувати умови навчання студентів та якість їх знань після закінчення навчального закладу. Інвестиції, які будуть вкладатися в вузи, будуть вигідними для роботодавців, які будуть отримувати спеціалістів, які підходять під потреби компанії і вузам, які отримають фінансову підтримку не тільки від держави.

Якщо вузи будуть тісно співпрацювати з компаніями, то вони підвищать свій рейтинг за рахунок того, що підвищиться кількість працевлаштованих студентів після закінчення вузу та навіть, під час навчання на останніх курсах [7].

Після закінчення вищих навчальних закладів студенту необхідно ще вкладати кошти в додаткові курси по програмуванню, які допоможуть знайти роботу. Одна з проблем, що в вузах не викладають необхідні предмети - відсутність викладацького

складу [8]. Якщо вузи будуть співпрацювати з компаніями, то буде можливість залучати практикуючих спеціалістів для проведення спеціальних курсів. Вони дадуть змогу студентам отримати ті навички та знання, які необхідні ринку інформаційних технологій.

Ще однією значною перевагою для обох сторін стане можливість залучати молодих спеціалістів до реальних проектів [9]. Компанії отримають спеціаліста з потрібними знаннями, студент отримає можливість попрактикувати отримані теоретичні знання, вуз підвищує свій рейтинг і можливість отримувати більше інвестицій.

Роботодавці ще наголошують на проблемі відсутності у студентів загального розуміння бізнесу та своєї ролі в проекті [10]. Спеціаліст в сфері інформаційних технологій має розуміти проект в цілому і бути готовим зробити не лише свою частину роботи, але й взяти відповідальність за інші процеси [11]. Цю проблему може вирішити залучення студентів в бізнес процеси компаній ще на стадії навчання.

Роботодавці не хочуть співпрацювати з людьми, які тільки закінчили навчання оскільки знають, що їх знань недостатньо для негайного включення в роботу [12]. Університетам варто скорегувати свої навчальні програми під потреби нинішнього ринку інформаційних технологій.

Існує декілька об'єктивних факторів неможливості працевлаштування випускника вищого навчального закладу:

- Відсутність розробленої системи взаємодії між вузом та роботодавцями;
- Різниця між потребами ринку інформаційних технологій та програми підготовки спеціалістів в вузах;
- Відсутність забезпечення якісних місця для проходження стажування.

Отже система корегування навчального плану в залежності від потреб ринку допоможе вирішити ряд проблем на державному рівні.

1. ОБГРУНТУВАННЯ ПОТРЕБИ СТВОРЕННЯ СИСТЕМИ КОРЕГУВАННЯ НАВЧАЛЬНОГО ПЛАНУ В ЗАЛЕЖНОСТІ ВІД ПОТРЕБ РИНКУ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Ми живемо в епоху настільки швидких змін, що нам часто не вистачає гнучкості для адаптації до них. Ця проблема характерна для всіх сфер і галузей: наприклад, ринку інформаційних технологій не вистачає самоорганізованих спеціалістів з знаннями [13] особливостей бізнесу і специфічних технологій, здатних оперативно реагувати на виклики сучасності. Зміни в індустріальному сегменті диктують попит на фахівців в тих галузях, які ще 5-7 років тому не існували зовсім. Освітня сфера не встигає оперативно трансформуватися, і це проблема світового масштабу, яку обговорюють на всіх профільних конференціях і форумах, намагаючись знайти шляхи вирішення [14].

В Україні глобальна освітня криза має національну специфіку, обумовлену економічними і політичними факторами [15]. Пріоритетна проблема української освіти - через недостатнє фінансування страждає матеріально-технічна база вузів. Для вирішення цього питання необхідно знайти зацікавлені компанії, які будуть готові платити за навчання студентів у вузах та надавати спонсорську допомогу [16]. Компанії не мають зацікавленості у студентах вищих навчальних закладів через недостатність або нерелевантність знань випускників. Саме тому актуальною є задача розробки програмного продукту корегування навчального плану в залежності від потреб ринку Інформаційних технологій. Для вирішення цієї проблеми необхідно створити систему, яка допоможе вузам країни корегувати свої навчальні плани. Система дасть можливість проглядати відкриті вакансії в компаніях в сфері інформаційних технологій. Проглядати навчальні плани вузів і проглядати порівняльний аналіз потреб ринку і знань та навиків, які надає вуз випускнику.

Мета дослідження полягає в розробці програмного продукту, який дозволить швидко та легко формувати дані для підтримки прийняття рішень з вирішення питань щодо корегування навчального плану в залежності від потреб ринку інформаційних технологій.

Програмний продукт повинен забезпечити наступні можливості:

- додавання, оновлення та видалення навчальних планів з існуючої бази даних;
- перегляд статистики по потребах ринку інформаційних технологій;
- порівняння навчальних планів з існуючими потребами ринку;
- можливість перегляду відкритих вакансій;
- перегляд профілю випускника вузу.

1.1 Огляд існуючих систем моніторингу ринку ІТ

Всі існуючі системи не дають можливості робити порівняльний аналіз кандидат - вакансія. В нашому випадку це вакансія - знання студенту вузу. Знизу представлен

З ростом кількості персоналу збільшується і щоденна кількість дій, які необхідно провести HR. Для спрощення пошуку співробітників і подальшого їх адміністрування створено безліч платформ. Було проведено дослідження з порівняння функціональних характеристик існуючих систем [17].

В огляді схожих систем було обрано десять найпопулярніших веб-рішень, якими користуються працівники відділу кадрів. Нижче короткий огляд кожної з них.

1. Greenhouse

Допомагає кандидатам отримати позитивний досвід користувачів, оптимізуючи процес найму. Підбір та проведення інтерв'ю з кандидатами і їх відбір обробляється так, щоб надати HR відділу максимальну кількість даних про кандидата і проводити більш точні інтерв'ю.

Платформа має, мабуть, самий повний функціонал, але виглядає при цьому досить складно і застарій. Робота з системою багато радості не принесе, а вартість 10000 \$ робить її не такою привабливою.

2. TalentScan.pro

TalentScan.pro - платформа повного циклу швидкого пошуку кадрів за кількома джерелами одночасно (LinkedIn, TurboHiring, GitHub, StackOverflow, Rabota.ua, Work.ua). Штучний інтелект платформи підбирає кандидатів автоматично на підставі переваг рекрутера і дозволяє відразу відправляти здобувачеві актуальна пропозиція, не виходячи з платформи. Як не дивно, цей продукт створений в Україні.

Платформа має підлаштовуватися під маркетингові кампанії по вакансіях і самонавчальний штучний інтелект, який дає правильні підказки. Платформа здатна значно підвищити продуктивність рекрутера, а також заощадити гроші на LinkedIn Recruiter пакет і на доступ до баз резюме на сайтах пошуку роботи.

3. Jobvite

Продукт для найму і відстеження кандидатів з декількома модулями. Jobvite Refer відкриває пошук здобувача в соціальних мережах, Jobvite Engage включає можливість створити пул запасних кандидатів, Jobvite Hire відстежує цих кандидатів, а за допомогою Jobvite Video можна організувати інтерв'ю по відеозв'язку.

На жаль сервіс має досить складну настройку ATS, і займає більше часу, ніж це насправді має бути при старті роботи з платформою. Однак в подальшому використанні платформа досить проста.

4. Zoho Recruit

Майданчик з системою CRM для найму співробітників. Має в своєму функціоналі систему відстеження кандидатів, інтеграція платформи для найму та управління співробітниками з CRM, системою відстеження здобувачів, інтеграцією з Microsoft Outlook і Google Apps. Крім цього даний продукт має програму структурування резюме і можливість sms-розсилок для спрощення комунікації з кандидатами.

Одне з головних переваг системи - низька вартість, при досить хорошому функціоналі. Але недолік - дуже незручний і не ефективний дизайн. Для рекрутерів це буде дуже важливо, адже працювати з платформою їм доведеться щодня.

5. SmartRecruiters

Дана платформа дає можливість просування HR-бренду компаній і пропонує «спільний найм». Продукт дозволяє за допомогою маркетингу здійснювати багатоканальний пошук претендентів, відбір претендентів і відправку пропозицій. Для того щоб завжди залишатися на зв'язку, співробітникам HR пропонується мобільний додаток.

Система більш оптимізована для великих компаній і мало підходить для невеликих бізнесів.

6. iCIMS Recruit

iCIMS Recruit - помічник відстеження вакансій і база даних, які дозволяють перейти від паперового до електронного документообігу. Сервіс дозволяє систематизувати робочий процес і комунікацію через соціальні мережі. Функціонал iCIMS можна поліпшити за допомогою модулів для соціальних мереж, а також модуля адаптації нових співробітників.

iCIMS Connect призначений для підтримки кандидатів з методами автоматизації маркетингу і надає інструменти для нових співробітників і рекрутерів, таких як електронні документи та управління завданнями.

7. CleverStaff

CleverStaff - продукт, який не ганяється за всім функціоналом конкурентів, а шукає свою нішу. Ви можете шукати нових кандидатів і мати єдину базу кандидатів з LinkedIn, пошти і сайтів пошуку роботи. Сервіс досить зручний для користування зі зрозумілим інтерфейсом. Має місцеву локальну підтримку на рідній мові.

8. UltiPro

UltiPro - платформа, яка призначена для залучення працівників, перерахування заробітних плат, нарахування премій, звітності, підрахунку KPI співробітників, а також для планування часу робочого дня і відпустки.

Платформа допомагає трансформувати вашу організацію для оптимізації роботи HR, проте погано пристосована для пошуку і роботи з новими співробітниками. Не має легкого пошуку та додавання нових співробітників. Також не передбачена можливість редагувати співробітників.

9. ClearCompany

Система пошуку фахівців з можливістю постановки цілей, і управління ефективністю. Завдяки своїм інструментам ClearCompany дозволяє HR-менеджеру поліпшити якість найму нових співробітників. Платформа полегшує завдання становлення кандидата в повноцінного командного учасника.

Програмне забезпечення має досить обмежений функціонал, але порівняно зручний у використанні.

10. ADP Workforce Now

ADP найбільше відома в корпоративних колах як рішення по роботі з зарплатами і компенсаціями. Також у компанії є свій HR-продукт - рішення, яке призначене для роботи зі штатом понад 50 співробітників.

Звичайно корисних показників оцінювання функціональних характеристик набагато більше, неможливо описати всі. Але це основні, які можуть допомогти вам з вибором платформи для HR [18].

Це може допомогти нашим дорогим колегам HR зберегти дорогоцінний час, який було витрачено на пошук таких платформ і аналіз цих продуктів.

Отже, отримаємо порівняльну таблицю (таблиця 1.1) характеристик вищезазначених систем за наступними критеріями. Критерії, за якими оцінювалися платформи наступні:

- інтеграція з LinkedIn - зручний інтерфейс перенесення профілів з LinkedIn;
- CRM - налаштовується система відправки листів кандидатам, включаючи пасивним, відстеження відповідей;
- карта навиків - настройка списку Скіл і компетенцій для вакансії і виставлення оцінки за підсумками інтерв'ю;

Таблиця 1.1(продовження) — Порівняльна таблиця існуючих систем для підбору персоналу

Карта навиків	+	+	+	-	-	-	-	-	-	-
Конфігурація етапів інтерв'ю	+	-	-	-	-	+	+	+	-	-
Мобільний додаток	+	-	+	+	+	+	+	+	-	+
Затвердження нових вакансій	+	-	+	+	+	+	+	+	+	+
UX	10	6	6	7	4	4	2	3	9	6
Дошка об'яв	+	+	+	+	+	+	-	-	+	+
Сторінка компанії	+	+	+	+	+	+	+	+	-	-
Інтеграції з Google Calendar	+	+	+	+	+	-	-	-	+	-
Воронка рекрутмерта	+	+	+	+	+	+	-	-	-	-
Статистика	+	+	+	+	+	-	+	+	+	+
Підтвердження офферів	+	+	+	+	+	+	+	+	+	+
Масова розсилка	+	+	-	-	-	-	-	-	-	--
Пошук по резюме	+	+	-	-	+	+	-	+	+	-
Постінг в соціальних мережах	+	+	+	+	+	+	-	+	+	-
Розширення для Google Chrome	+	+	-	-	-	-	-	-	-	-

Отже, необхідно створити систему яка має більшість з цих характеристик та додати новий функціонал, який допоможе надавати рекомендації щодо корегування навчальних планів в залежності від потреб ринку інформаційних технологій. Важливо звернути увагу на простоту системи. Більшість з представлених систем

розраховані на великий і не ціленаправлений функціонал. Варто зосередитися на специфіку навчальних закладів та вимог роботодавців. Для цього необхідно розробити такі функції як оцінка важливості навиків для отримання роботи та алгоритм порівняння існуючих навчальних планів до навиків, які необхідно роботодавцям.

1.2 Вхідні та вихідні дані

Необхідними вхідними даними для коректної роботи програмного забезпечення є занесення інформації про існуючі вакансії, інформація про профіль студента після закінчення вузу.

В інформації про вакансію необхідно внести дані про назву вакансії, назва компанії замовника, опис проекту, навички необхідні для заняття даної позиції. Для кожної навички, роботодавець має вказати степінь важливості. Кількість навичок є обмеженою для кожної вакансії і не мають містити персональні якості людини.

Профіль студента формується з назви вузу, назва напрямку підготовки, списку курсів, які читаються у вищому навчальному закладі, і списку навиків для кожного курсу, які вивчають студенти.

В якості вихідних даних користувач може отримати статистику по необхідним навикам для кожної вакансії, загальну статистику по популярним технологіям, порівняльний аналіз профілю випускника вузу та наявних вакансій на ринку. Ще програмний продукт має можливість підібрати найбільш вдалу вакансію для студента та показати, що необхідно вивчити додатково, що б підходити на тут чи іншу позицію повністю.

1.3 Висновки до розділу 1 та постановка задачі

Проаналізовано існуючі системи по підбору персоналу і виделені головні особливості таких веб рішень:

1. Можливість зберігання інформації про вакансії

2. Можливість зберігання профелів кандидатів
3. Наявність статистики за різними показниками
4. Організовано пошук по вакансіям та кандидатам

Сформульовані наступні завдання дослідження:

1. Проаналізувати можливості сучасних інформаційних систем для набору персоналу.
2. Проаналізувати методи порівняння існуючих потреб ринку та навиків випусника вузу
3. Розробити алгоритм оцінки важливості навиків для існуючих вакансій
4. Розробити алгоритм порівняння існуючих потреб ринку та навиків випускнику вищих навчальних закладів
5. Розробити веб застосунок для корегування навчальних планів в залежності від потреб ринку інформаційних технологій.
6. Спроекувати архітектуру системи орегування навчальних планів в залежності від потреб ринку інформаційних технологій
7. Розробити спроектовану систему.

2. ПРОЄКТУВАННЯ СИСТЕМИ КОРЕГУВАННЯ НАВЧАЛЬНОГО ПЛАНУ В ЗАЛЕЖНОСТІ ВІД ПОТРЕБ РИНКУ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Багато іт-компанії нарікають, що українські вузи випускають фахівців, які не відповідають вимогам сучасних технологічних компаній.

На даний момент українські вузи не випускають готових до роботи фахівців. Після закінчення вузу коли вже необхідно працевлаштуватися, отриманих в інституті знань не вистачає навіть для позиції Junior розробника.

В чому причина? Вузи перебувають на мізерному державному фінансуванні і у них немає можливості освоювати сучасні методи навчання. Також немає можливості досліджувати, що необхідно зараз ринку. Нема технічної бази для досліджень [19].

Також далеко не всі викладачі можуть і хочуть освоювати останні технологічні віяння, нові продукти, не кажучи вже про бізнес-складової українського і міжнародного ІТ-ринку [20]. У них немає досвіду роботи в бізнес-структурах, вони не можуть дати студентам необхідну бізнес-хватку, тому що самі їй не володіють.

Вузи відкриті для співробітництва, так як хочуть бути в тренді. Але вони відкриті найчастіше лише в рамках безкоштовних програм навчання [21]. Тобто якщо потенційний роботодавець зацікавлений, він може проводити для студентів вузу власний курс або читати лекції і витратити на це свій час і гроші, розраховуючи на те, що в майбутньому ці студенти прийдуть до нього працювати [22]. Але ми можемо заохотити університети самостійно надавати студентами необхідні знання і отримувати фінанси на це від роботодавців, які в майбутньому зможуть надати роботу випускникам.

Зараз механізм пошуку роботи після вузу такий: студент шукає роботу в Україні, щоб набратися досвіду, відвідує спеціалізовані курси, і вже після того як

отримує рівень middle, шукає можливості для того що б піти на роботу [23]. Це спричиняє три проблеми:

1. Не вистачає Junior спеціалістів
2. Після закінчення вузу ще необхідно витратити час на доосвіту.
3. Навчання в університеті вважається витраченим часом.

За дослідженнями, в цілому по ринку випускники готові на 30% і їм доводиться вчитися на роботі. Університет дає їм базові навички, вміння мислити, аналізувати, шукати інформацію, готувати презентації. Але, на жаль, студенти отримують професії, які в більшості своїй не затребувані.

В 2009 році веб-портал DOU.UA проводив опитування серед своїх відвідувачів, на скільки корисним було навчання в вузах. Виявилось, що цінних знань, які надає вуз всього 15% від всіх знань. Майже 40% зайняли недостатньо корисні знання. Десь такий самий відсоток мають знання, які більш менш підходять для подальшого працевлаштування студентів (рисунок 2.1).

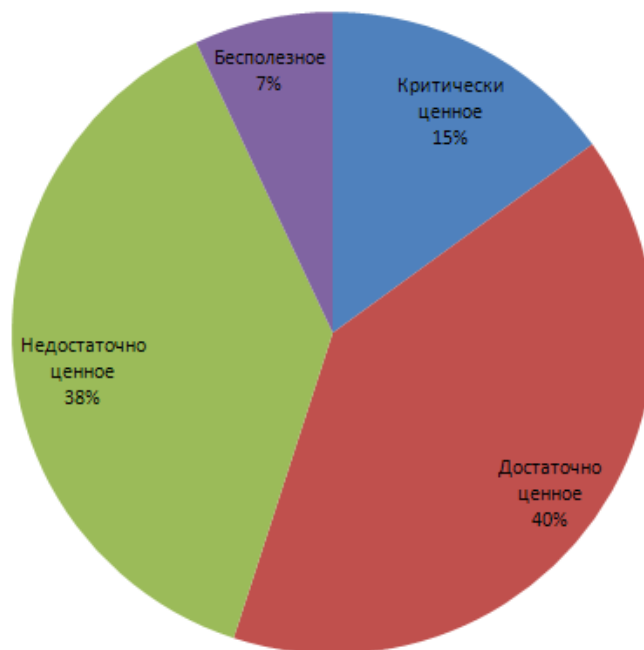


Рисунок 2.1— Діаграма показників корисності знань, які надає вуз

Але якщо, взяти не загальну статистику, а подивитися на конкретні вузи, то можна побачити, що знання, які необхідні для роботи не досягають навіть 30%. Т Також було кілька відповідей про те що не було перешкод, або були всі

перераховані вище перешкоди, багато нарікань на навчальну програму - про те що вона застаріла, відірвана від практики, багато непотрібних предметів і т. д.

Щоб змінити ситуацію, необхідно розробити систему корегування навчального плану.

Для того щоб створити таку систему необхідно створити постійно оновлювальну систему, яка буде через якийсь період часу збирати і опрацьовувати дані по відкритим вакансіям. На основі цієї інформації можна робити висновки про потреби ринку інформаційних технологій. Необхідну інформацію можна буде збирати з так званих Job-сайтів.

З іншого боку необхідно збирати інформацію по реальним програмам університетів. Тут постає проблема, що інформація не структурована по одному принципу. Виходом з цього є створення власної системи парсингу, яка з великої кількості тексту буде вибирати необхідну інформацію приводити її до стандартного вигляду.

Третім етапом розробка системи корегування навчального плану вищих навчальних закладів в залежності від потреб ринку Інформаційних технологій вимагає порівняння даних отриманих з JOB-сайтів та програми курсів в університетах. Складність цього етапу в правильно структурованих звітів.

Для реалізації даної системи необхідно було обрати базу даних, в якій буде зберігатися вся інформація. В ході досліджень було обрано базу даних MongoDB. Ця база даних є документо-орієнтована система керування базами даних(СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

Оскільки на даний момент найбільш зручний спосіб користування такими системами – у форматі онлайн застосунку то необхідно було розробити зручний інтерфейс користувача з використанням сучасних технологій для Front-End. Для цих цілей було обрано HTML, CSS, RAZOR. Для Back-

End використано .NET CORE. Після тестувань різних методів було виявлено, що .NET CORE найкраще працює з MS SQL EXPRESS.

2.1 Функціональні можливості системи

Перед тим як починати роботу на створення програмного продукту, була розроблена діаграма прецедентів (рисунок 2.2) . Було вирішено, що системою корегування навчальних планів в залежності від потреб ринку інформаційних технологій зможуть користуватися два головні актори: представник компанії та представник вищого навчального закладу. З діаграми прецедентів видно, що у двох вищезазначених акторів є як різні функціональні можливості, так і спільні.

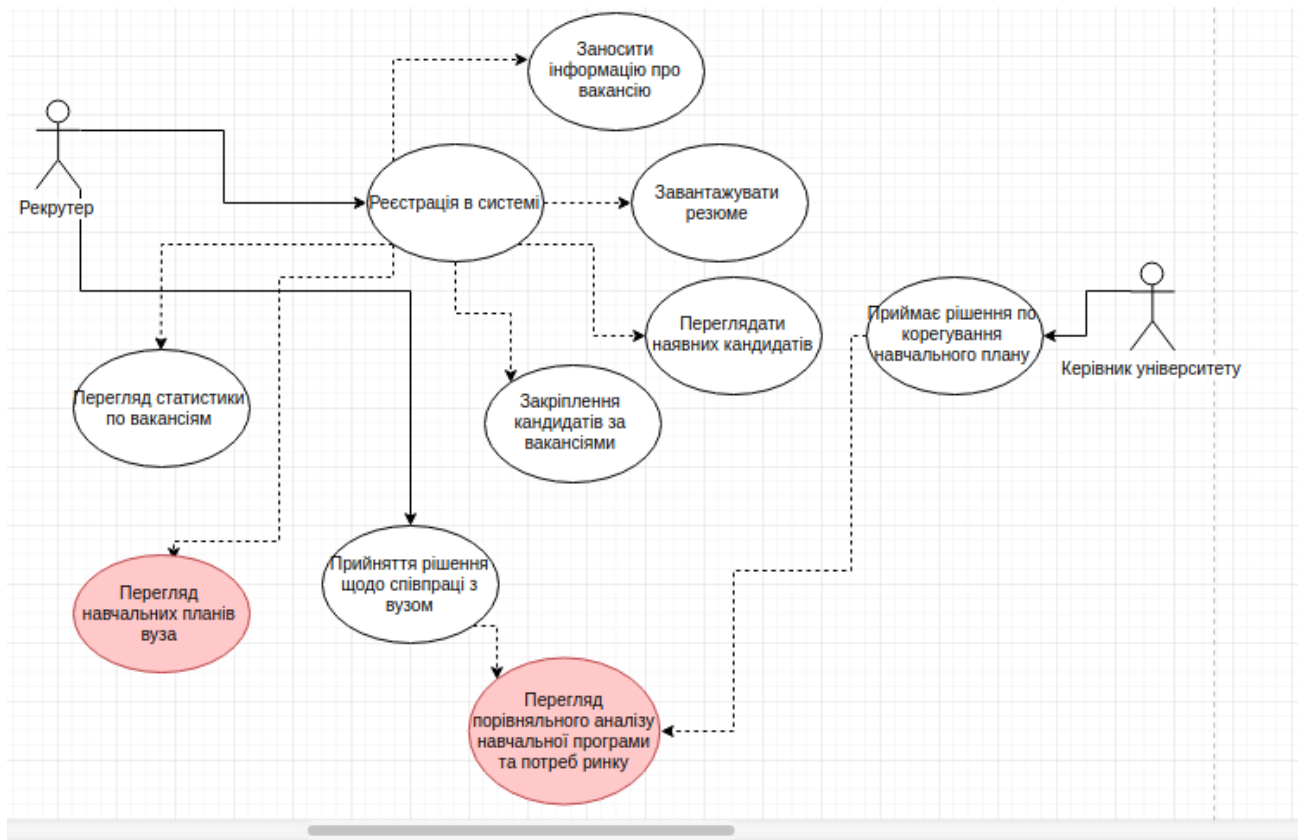


Рисунок 2.2 — Діаграма прецедентів створюваної системи

Розглянемо спільні можливості для обох акторів. Обидва мають обов'язково зареєструватися в системі. При реєстрації кожен користувач має обрати тип актора. Від цього в майбутньому буде залежати доступний йому функціонал. Також користувачі мають внести основну інформацію про себе: прізвище, ім'я, по батькові,

контактний номер телефону, e-mail, та пароль. Вся ця інформація є обов'язковою для проходження реєстрації і аутентифікації в подальшому. Обидва актори мають можливість переглядати статистику по необхідним навичкам та статистику ринку інформаційних технологій.

Розглянемо можливості яка надає система для користувача, який представляє компанію і набирає собі персонал. По-перше актор-рекрутер може створювати вакансії, вносити зміни в уже існуючі вакансії, видаляти вакансії. Передбачена можливість виставлення пріоритетності для кожної технології, яка необхідна для данної вакансії.

З боку представника вузу забезпечена функція створення сторінки університету, дисциплін, які включають певні навички, які студент отримує після закінчення навчання. Найголонішим є перегляд рекомендацій щодо корегування навчального плану в залежності від потреб ринку інформаційних технологій. Система показую на які вакансії може претендувати випускник вузу та рекомендує, які технології варто включити в програму вищого навчального закладу.

2.2 Сховища даних

В даний час однозначного визначення сховища даних не існує, тому що розроблено велику кількість різноманітних архітектур та технологій СД, а самі зберігачі використовуються для вирішення найрізноманітніших завдань. Кожен автор вкладає в це поняття своє видіння питання. Загальні вимоги можна визначити наступним визначенням СД, який не претендує на повноту та однозначність, але дозволяє зрозуміти основну ідею.

Сховище даних - різновид систем зберігання, орієнтований на підтримку процесу аналізу даних, що забезпечує цілісність і хронологію даних, а також високу швидкість виконання аналітичних запитів.

Найважливішим елементом СД є семантичний шар - механізм, що дозволяє аналізувати дані за допомогою бізнес-термінів предметної області. Семантичний

шар дає користувачеві можливість зосередитися на аналізі та не думати про механізми отримання даних.

Типове СД істотно відрізняється від звичайних систем зберігання даних. Головною відмінністю є цілі використання. Наприклад, реєстрація продажів та виписки відповідних документів - завдання рівня OLTP-систем, що використовують звичайні реляційні СКБД. Аналіз динаміки продажів і попиту за кілька років, що дозволяє розробляти стратегію розвитку фірми та планувати роботу з постачальниками та клієнтами, зручніше виконати за підтримки СД.

Друга важлива відмінність полягає в динаміці зміни даних. Бази даних в OLTP-системах характеризуються дуже високою динамікою зміни записів через повсякденну роботу великого числа користувачів [24] (відколи, до речі, велика вірогідність появи протиріч, помилок, порушень цілісності даних тощо). Що стосується СД, то дані з нього не видаляються, а поповнення відбувається відповідно до певного правила (раз на годину, день, тиждень, в певний час).

Отже, для системи корегування навчальних планів в залежності від потреб ринку необхідно вибрати правильне сховище даних. Необхідно буде зберігати інформацію про вакансії та інформацію про профіль студента після закінчення вищого навчального закладу. Для того щоб обрати правильну базу даних було проведено ряд досліджень.

При виборі системи зберігання даних необхідно було звертати увагу на легкість інтегрування з обраною мовою програмування. Необхідно було звертати увагу на масштабованість системи. Оскільки вакансій зараз дуже багато і потреба в спеціалістах в цій сфері тільки розширюється, то необхідно зберігати велику кількість інформації про відкриті вакансії. З іншого боку необхідно зберігати інформацію про вузи і спеціалізації, які є в них. Для кожної спеціалізації важливо зберігати навички, які студент отримає після закінчення вищого навчального закладу.

Враховували, що обрана система зберігання даних не має займати забагато місця і забезпечувати швидкодію. Оскільки розроблена система є веб застосунком, це накладає певні обмеження на об'єми інформації, яка оброблюється і на скільки

швидко це відбувається. Важливо що б користувач не відчував затримок між запитом і реагуванням системи.

2.3 Алгоритм оцінювання необхідності навиків

В процесі досліджень було виявлено що не всі технології рівноцінні і мають однакову цінність для вакансій. Різні технології необхідні для різних вакансій. Іноді буває, що одні й ті самі навички для різних вакансій мають різний пріоритет. За основу оцінки необхідних навиків була взята система оцінювання, яку висунули компанії в сфері інформаційних технологій.

ІТ-компанії показали єдині вимоги до Джуніор позицій. Асоціація «ІТ України» представила уніфіковані рекомендації української ІТ-індустрії до junior-фахівцям - тестувальникам і розробникам. Як пояснюють в асоціації, цей документ допоможе тим, хто хоче почати будувати кар'єру в галузі і зрозуміти, які навички та вміння потрібні в першу чергу. До компаній, які брали участь в данному процесі брали участь такі всесвітньо відомі компанії, як Astound Commerce, EPAM, Cicklum, GlobalLogic, Innovecs, Luxoft, SoftServe, Terrasoft. Вони пропонують розділяти навички за ступінь знань: початківець, початківець/опціонально, середній рівень, високий рівень.

Цей список був складений для Front-End розробників, Back-End розробників двох напрямків: Java та .NET, тестувальників. Але оскільки на даний час є набагато більше спеціальностей та вакансій, то необхідно було передбачити у системі можливість робити рекомендації на основі не певного списку вимог, а власне з існуючих вакансій.

В системі передбачена можливість вносити у вакансії необхідні навички та оцінювати їх важливість. Після цього система зможе більш коректно робити рекомендації, щодо знань, які необхідно додати в навчальний план вищих учбових закладів.

Необхідно було зрозуміти, яким чином буде визначатися необхідність знань. Було вибрано оціночну систему. При занесенні інформації про необхідні навички

для певної позиції, сама компанія, яка відкриває позицію, визначає ваажливість знань технологій та їх кількість.

Це було досягнуто введенням системи оцінок від 1 до 5, Відповідно 1 - менш важливий навик, 5 - дуже важливо, для отримання роботи. В подальшому кожна оцінка переводиться в коефіцієнт важливості за формулою:

$$K_{\text{важ}} = \frac{100\% * P_i}{\sum P}$$

, де $K_{\text{важ}}$ – коефіцієнт важливості навика, P_i – оцінка, яку виставив користувач даному навику, $\sum P$ – сума всіх виставлених оцінок для навиків, користувачем.

В такому випадку кожний навик отримає свою вагу по відношенню до інших. Далі проводиться оцінка кожного навика кандидата в залежності від вагів оцінки. Проводиться сумування набраних вагів тільки тих, які присутні у профілі навиків випускника вищого навчального закладу. Отримаємо рейтингову оцінку якості та необхідності знань кандидата для певних вакансій.

Ця система вводиться для того що б отримати виставлену оцінку важливості у відсотковому відношенні до інших оцінок. Це дозволить правильно оцінити на скільки дисципліни, які читаються в вузі відповідають нинішнім потребам ринку.

Цю систему оцінювання також використали при створенні статистики, яку можуть побачити користувачі системи. За допомогою статистики можна побачити список навичок необхідних для того, що б зайняти певну позицію. Тако ж система дасть рекомендації, яких знань не вистачає для того, що б роботодавець прийняв на роботу студента після закінчення вузу.

2.4 Алгоритм порівняння потреб ринку та існуючих навиків

Отже основною проблемою розробки системи корегування навчального плану в залежності від потреб ринку інформаційних технологій стало створення алгоритму пошуку навичок, які необхідні для вакансії та існуючі знання у студента.

Схожі системи не мають такої опції. Необхідно було продумати, який саме алгоритм буде найшвидший і найдієвіший у випадку розробленні даної системи.

Необхідно було врахувати швидкодію методу, простоту в реалізації та можливість адаптованості під нові умови. Були використані регулярні вирази для задання шаблонів пошуку.

Необхідно провести структурний аналіз набору компетенцій, які необхідні для отримання роботи та набору навиків випускника вузу. Проводимо порівняльний аналіз структури знань, яку пропонує вуз по відношенню до компетенцій, яку визначають роботодавці. Для цього розробимо алгоритм порівняння (рисунок 2.3).

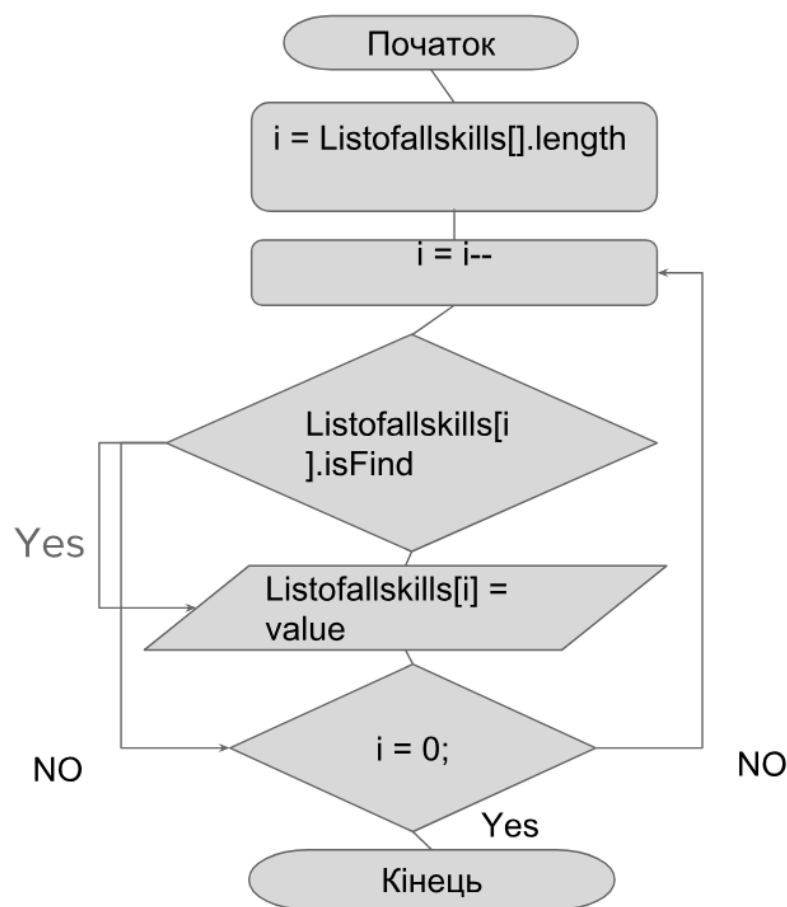


Рисунок 2.3 - Діаграма прецедентів створюваної системи

На вході в алгоритм ми маємо колекцію зі всіх навиків для вакансій, які зараз існують в системі. В системі передбаченне зберігання навиків з їх коефіцієнтом важливості і приналежність навика до певної вакансії.

Алгоритм перевіряє по черзі кожен з наявних навиків, на предмет існування його в навчальному плані.

В ході парсингу документу алгоритм знаходить схожі навик за допомогою порівняння і використання регулярних виразів. У випадку знаходження співпадіння, додається коефіцієнт, який обраховується алгоритмом необхідності навику

Алгоритм завершує свою роботу, коли останній навик зі списку був оброблений.

Після закінчення роботи алгоритму користувач може побачити відсоткове відношення до трьох найбільш підходящих вакансій. Також система покаже навик, які не знайшла в навчальному плані, але які необхідні, що б зайняти позицію.

2.5 Регулярні вирази

Регулярні вирази (regular expressions, RegExp) - набори символів, що застосовуються для пошуку текстових рядків, відповідних необхідним умовам. Результатом застосування регулярного виразу є підмножина даних, відібране згідно з логікою, закладеної в вираженні [25]. Регулярні вирази застосовуються в будь-яких завданнях з пошуку в множині даних, для яких потрібно отримувати вижимки за певними правилами.

Класи `StringBuilder` і `String` надають достатню функціональність для роботи з рядками. Однак `.NET` пропонує ще один потужний інструмент - регулярні вирази. Регулярні вислови ефективний і гнучкий метод по обробці великих текстів, дозволяючи в той же час істотно зменшити обсяги коду в порівнянні з використанням стандартних операцій з рядками.

Основна функціональність регулярних виразів в `.NET` зосереджена в просторі імен `System.Text.RegularExpressions`. А центральним класом при роботі з регулярними виразами є клас `Regex`. Наприклад, у нас є певний текст і нам треба знайти в ньому все словоформи якогось слова.

Більшість символів в регулярних виразах представляють самі себе, за винятком групи спеціальних символів «`[] \ / ^ $. | ? * + () {}`». Якщо ці символи потрібно представити в якості символів тексту, їх слід екранувати зворотною косою межею «`\`».

Якщо ці спецсимволи зустрічаються без зворотної косої межі, значить у них особливі значення в регулярних виразах:

- «[^]» - каретка, циркумфлекс або просто галочка. Початок рядка;
- «\$» - знак долара. Кінець рядка;
- "." - крапка. Будь-який символ;
- «*» - знак множення, зірочка. Будь-яка кількість попередніх символів;
- «+» - плюс один або більше попередніх символів;
- "?" - знак запитання. 0 або 1 попередніх символів;
- "(" - круглі дужки. Угруповання конструкцій;
- «|» - вертикальна лінія. Оператор «АБО»;
- "[]" - квадратні дужки. Будь-який з перелічених символів, діапазон.

Якщо перший символ в цій конструкції - «[^]», то масив працює навпаки – символ, що перевіряється, не повинен збігатися з тим, що перераховано в дужках;

- "{ }" - фігурні дужки. Повторення символу кілька разів;
- «\» - зворотний слеш. Екранування службових символів.

Також існують спеціальні метасимволи, ними можна замінити деякі готові конструкції:

- \b - позначає не символ, а кордон між символами;
- \d - цифровий символ;
- \D - нецифровий символ;
- \s - символ пробілу;
- \S - непробельний символ;
- \w – літерний символ;
- \W - будь-який символ. Виключенням є підкреслення, цифровий та літерні символи.

літерні символи.

Перемінна модель задає регулярне вираження для перевірки адреси електронної пошти. Дане вираз пропонує нам Microsoft на сторінках msdn.

Для перевірки відповідності строку шаблону використовується метод `IsMatch: Regex.IsMatch` (електронна пошта, шаблон, `RegexOptions.IgnoreCase`).

2.6 Висновки до розділу 2

В другому розділі були розглянуті проблеми, які спричинили виникнення потреби розроблення системи корегування навчального плану в залежності від потреб ринку інформаційних технологій. Обґрунтували потреби розроблення вищезазначеної системи.

У другому розділі описали функціональні можливості системи корегування навчального плану в залежності потреб ринку інформаційних технологій. Виділений основний функціонал та головні актори (користувачі) системи.

Були запропоновані алгоритми для порівняння потреб ринку та існуючих навиків у випусника вузу та обрахунку коефіцієнта важливості навиків, для отримання вакансії та успішного працевлаштування.

В другому розділі розглянули варіанти збегігання даних та визначилися з основною базою даних. Визначили основні критерії за якими необхідно обирати БД для системи корегування навчальних планів в залежності від потреб ринку інформаційних технологій

3. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Для реалізації поставленої задачі треба обрати найбільш оптимальний шлях для вирішення поставлених задач. У ході роботи над дипломним проектом було проведено аналіз, який допоміг обрати методи програмної реалізації для кращого виконання роботи.

3.1 Засоби розробки

Для створення веб-ресурсу була використана мова програмування C# та фреймворк .NET CORE для створення back-end частини. Для розробки інтерфейсу використана мова розмітки HTML та мова програмування JavaScript. Була використана технологія Razor для впровадження в веб-сторінки серверного коду.

3.1.1. Основна архітектура проекту. Паттерн MVC

В якості основи архітектурного рішення було взято паттерн MVC (Model, View, Controller). Це многорівневе архітектурне рішення, яке дозволяє розділити проект на логічні рівні. Шаблон проектування MVC передбачає поділ даних програми, призначеного для користувача інтерфейсу і керуючої логіки на три окремих компоненти: модель, представлення і контролер [26] - таким чином, що модифікація кожного компонента може здійснюватися незалежно. Відношення між компонентами паттерна можна описати схемою представленою на рисунку 3.1.

Презентаційний шар (рівень представлення): це той рівень, з яким безпосередньо взаємодіє користувач. Цей рівень включає компоненти користувацького інтерфейсу, механізм отримання введення від користувача. Применительно до asp.net mvc на даному рівні розташовані представлення і всі ті компоненти, які складають користувацький інтерфейс [27] (стилі, статичні сторінки html, javascript), а також моделі представлень, контролери, об'єкти контексту запиту.

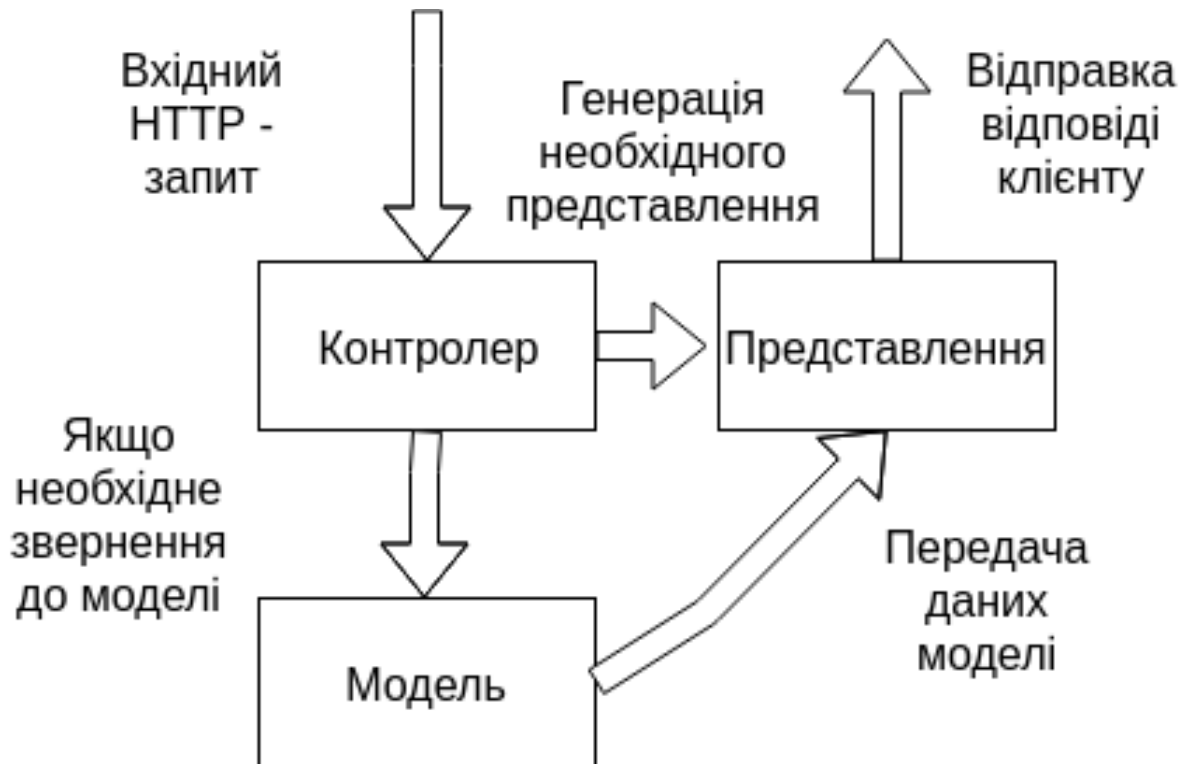


Рисунок 3.1— Загальне схема шаблону проектування MVC

Бізнес-рівень (рівень бізнес-логіки): містить набір компонентів, які відповідають за обробку отриманих від рівня представлень даних, реалізує всю необхідну логіку програми, всі підрахунки, взаємодіє з базою даних і передає рівень представлення результату обробки [28].

Шаблон доступу до даних (рівень доступу до даного) (рисунок 3.2): зберігає моделі, описуючи використовувані сутності, також розміщуються специфічні класи для роботи з різними технологіями доступу до даними, наприклад, клас контексту даних Entity Framework. Тут також зберігаються репозиторії, через які рівень бізнес-логіки взаємодіє з базою даних.

Таке розмежування компонентів додатків дозволяє реалізувати концепцію розподілу відповідальності, при якій кожен компонент відповідає за свою строго вичерпну сферу. В зв'язку з чим легше побудувати роботу над окремими компонентами. І благодяки цьому додатку легше розробляти, підтримувати та тестувати окремі компоненти. Допустим, якщо нам важлива візуальна частина або

фронтенд, то ми можемо тестувати представлення незалежно від контролера. Можливо, ми можемо зосередитися на беккенде та тестувати контроллер.

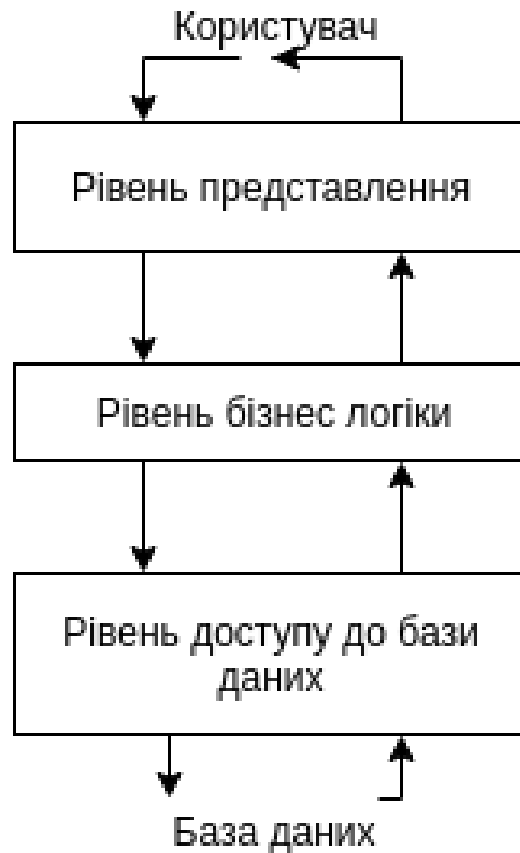


Рисунок 3.2 — Загальне схема рівнів шаблону проектування MVC

Архітектура MVC має ряд переваг:

1. Розширюваність

Інфраструктура MVC Framework побудована у вигляді набору незалежних компонентів, які задовольняють інтерфейс .NET або створені на основі абстрактного базового класу [29]. Компоненти, подібні системі маршрутизації, механізму візуалізації та фабрики контролерів, можна легко замінити іншими компонентами з власною реалізацією. В загальному випадку для кожної компоненти MVC Framework пропонує три можливості:

- Використання стандартної реалізації компонента в тому вигляді, як вона є (це має бути достатньо для більшості додатків);
- Створення підкласу з стандартної реалізації з метою корекції існуючого поведінки;

— Повна заміна компонента нової реалізації інтерфейсу або абстрактного базового класу;

2. Жорсткий контроль над HTML і HTTP

Інфраструктура ASP.NET MVC генерує ясний та відповідний стандартам код розмітки. Вбудовані допоміжні методи HTML виготовляють відповідні стандарти виведення, але існує також значно більше значима філософська зміна порівняно з веб-формами [30]. Замість генерації величезного обсягу важко піддається керуванню HTML-розмітки інфраструктури MVC Framework стимулює створення простих та елегантних елементів, оформлених стилями CSS.

Згенеровані ASP.NET MVC Сторінки не містять ніяких даних View State, тому вони менше типових сторінок ASP.NET Web Forms за розміром [29]. Незважаючи на сучасні Швидкі зв'язки, така економія трафіку як і раніше підвищує комфорт кінцевого користувача та допомагає скоротити витрати, пов'язані з запуском популярних веб-додатків.

3. Легке тестування

Природний розподіл різних відповідальностей додатки з незалежних один від одного частин програмного забезпечення, яке підтримується архітектурою MVC, дозволяє спочатку будувати легко супроводжувані і тестовані програми. Проте проектувальники ASP.NET MVC на цьому не зупинилися. Для кожного фрагмента компонентно-орієнтованого проекту інфраструктури вони забезпечили структурованість, необхідну для задоволення вимог модульного тестування і засобів імітації.

4. Потужна система маршрутизації

Удосконалення технології побудови веб-додатків супроводжувалося розвитком стилю URL-адрес. Адреси URL, подібні до наведеного нижче:

`/App_v2/MyShop/Page.aspx?action=search&type=ball&id=2106&color=silver`

зустрічаються все рідше, а на зміну їм приходить більш простий і зрозумілий формат такого вигляду:

`/shop/ball/2106-silver`

Існує ряд вагомих причин для того, щоб піклуватися про структуру URL-адрес.

По-перше, пошукові механізми надають такі ключові слова, що містяться в URL, більшу вагу.

По-друге, багато веб-користувачі досить кмітливі, щоб зрозуміти URL, і цінують можливість здійснення навігації шляхом введення запиту в адресному рядку свого браузера.

По-третє, коли хтось розуміє структуру URL-адреси, вони з більшою ймовірністю пройдуть по ньому, поділяться з іншими або навіть зачитають його вголос по телефону.

По-четверте, при такому підході в Інтернеті не розкриваються технічні нюанси, структура каталогів і імен файлів програми; отже, ви вільні змінювати лежить в основі сайту реалізацію, не порушуючи працездатності всіх вхідних посилань.

5. Сучасний API-інтерфейс

Платформа Microsoft .NET розвивалася з кожним великим випуском, підтримуючи - і навіть визначаючи - багато передових аспекти сучасного програмування.

Версія ASP.NET MVC 5 побудована для .NET Framework 4.5.1, тому її API-інтерфейс може в повній мірі задіяти останні нововведення мови і виконуючого середовища, в тому числі ключове слово `await`, що розширюють методи, лямбда-вирази, анонімні і динамічні типи, а також мову інтегрованих запитів (Language Integrated Query - LINQ) [31]. Багато методи і шаблони кодування API-інтерфейсу MVC Framework слідує більш чіткої та виразної композиції, ніж це було можливо в ранніх платформах.

6. Інфраструктура ASP.NET MVC має відкритий код

На відміну від попередніх платформ веб-розробки виробництва Microsoft, початковий вихідний код ASP.NET MVC доступний для вільного завантаження і навіть для модифікації і компіляції з метою отримання власної версії цієї інфраструктури. Це буквально неоціненно при налагодженні коду, що звертається до системного компоненту, коли потрібно крок за кроком виконати його код (і навіть ознайомитися з коментарями програмістів, які написали цей код). Це також

корисно, якщо ви створюєте вдосконалений компонент і хочете бачити, які існують можливості розробки, або дізнатися, як дійсно працюють вбудовані компоненти.

До того ж згадана можливість зручна і тоді, коли не влаштовує робота того чи іншого компонента, коли потрібно знайти помилку або коли необхідно отримати доступ до того, що недоступно за допомогою інших засобів - цікавий для компонент можна просто змінити самостійно. Однак при цьому доведеться відстежувати свої зміни і повторно їх застосовувати при модернізації до більш нової версії інфраструктури

Вихідний код ASP.NET MVC поширюється відповідно до відкритої ліцензією Microsoft (Microsoft Public License - Ms-PL) - ліцензією програмного забезпечення з відкритим кодом, схваленої Ініціативою з підтримки ПЗ з відкритим кодом (Open Source Initiative - OSI). Це означає, що вихідний код можна змінювати, розгортати і навіть поширювати у вигляді похідного проекту. Вихідний код MVC доступний для завантаження за адресою ASP.NET MVC Source Code [32].

3.1.2. Фреймворк .NET CORE

.NET Core - це універсальна платформа розробки з відкритим кодом, яку підтримує корпорація Майкрософт і співтовариство .NET на сайті GitHub. Вона є кроссплатформенною (підтримує Windows, macOS і Linux) і може використовуватися для створення додатків для пристроїв, хмари і Інтернету речей.

З одного боку, ASP.NET Core є продовженням розвитку платформи ASP.NET. Але з іншого боку, це не просто черговий реліз. Вихід ASP.NET Core фактично означає революцію всієї платформи, її якісна зміна [33].

Розробка над платформою почалася ще в 2014 році. Тоді платформа умовно називалася ASP.NET vNext [35]. У червні 2016 року вийшов перший реліз платформи. А в травні 2018 року побачила версія ASP.NET Core 2.1, яка власне і охоплена в поточному керівництві.

ASP.NET Core тепер повністю є opensource-фреймворком. Всі вихідні файли фреймворку доступні на GitHub.

ASP.NET Core може працювати поверх крос-платформної середовища .NET Core, яка може бути розгорнута на основних популярних операційних системах: Windows, Mac OS X, Linux. І таким чином, за допомогою ASP.NET Core ми можемо створювати крос-платформні додатки. І хоча Windows як середовище для розробки і розгортання програми досі превалює, але тепер вже ми не обмежені тільки цією операційною системою. Тобто ми можемо запускати веб-додатки не тільки на ОС Windows, але і на Linux і Mac OS. А для розгортання веб-додатки можна використовувати традиційний IIS, або крос-платформний веб-сервер Kestrel [36].

Хоча ASP.NET Core переважно націлене на використання .NET Core, але фреймворк також може працювати і з повною версією фреймворка .NET.

Завдяки модульності фреймворка всі необхідні компоненти веб-додатки можуть завантажуватися як окремі модулі через пакетний менеджер Nuget. Крім того, на відміну від попередніх версій платформи немає необхідності використовувати бібліотеку System.Web.dll.

ASP.NET Core включає в себе фреймворк MVC, який об'єднує функціональність MVC, Web API і Web Pages. У попередніх версії платформи дані технології реалізувалися окремо і тому містили багато дублюючої функціональності. Зараз же вони об'єднані в одну програмну модель ASP.NET Core MVC. А Web Forms повністю пішли в минуле.

Крім об'єднання вищезазначених технологій в одну модель в MVC був доданий ряд додаткових функцій.

Однією з таких функцій є тег-хелпери (tag helper), які дозволяють більш органічно поєднувати синтаксис html з кодом C #.

ASP.NET Core характеризується розширюваністю. Фреймворк побудований з набору щодо незалежних компонентів. І ми можемо або використовувати вбудовану реалізацію цих компонентів, або розширити їх за допомогою механізму спадкування, або зовсім створити і застосовувати свої компоненти зі своїм функціоналом.

Також було спрощено управління залежностями і конфігурація проекту. Фреймворк тепер має свій легкий контейнер для впровадження залежностей, і

більше немає необхідності застосовувати сторонні контейнери, такі як Autofac, Ninject. Хоча при бажанні їх також можна продовжувати використовувати.

В якості інструментарію розробки ми можемо використовувати останні випуски Visual Studio, починаючи з версії Visual Studio 2015. Крім того, ми можемо створювати додатки в середовищі Visual Studio Code, яка є крос-платформної і може працювати як на Windows, так і на Mac OS X і Linux.

Для обробки запитів тепер використовується новий конвеєр HTTP, який заснований на компонентах Katana і специфікації OWIN. А його модульність дозволяє легко додати свої власні компоненти.

Якщо підсумувати, то можна виділити наступні ключові відмінності ASP.NET Core від попередніх версій ASP.NET:

- новий легкий і модульний конвеєр HTTP-запитів;
- можливість розгорнути додаток як на IIS, так і в рамках свого власного процесу;
- використання платформи .NET Core і її функціональності;
- поширення пакетів платформи через NuGet;
- інтегрована підтримка для створення та використання пакетів NuGet;
- єдиний стек веб-розробки, що поєднує Web UI і Web API;
- конфігурація для спрощеного використання в хмарі;
- вбудована підтримка для впровадження залежностей;
- можливість розширення;
- кросплатформність: можливість розробки і розгортання додатків ASP.NET на Windows, Mac і Linux;
- розвиток як open source, відкритість до змін.

3.1.3 Мова розмітки HTML

Термін HTML (HyperText Markup Language) перекладається на українську як «Мова маніпулювання гіпертекстами». Ключовим тут є слово «гіпертекст». Під ним мається на увазі, на відміну від простого тексту, можливість створення повноцінних

документів, що містять різноманітні шрифти, графіку, звук, що досягається шляхом включення в текст спеціальних вказівок (їх іноді називають «тегами»). Структура тегів чітко обумовлена в описі мови .

Видача гіпертекстового документа (або, як часто говорять, «сторінки») на екран або папір проводиться відповідно до цих вказівок. Проте в переважній більшості випадків автор документу не обумовлює багато деталей, вважаючи, що вони некритичні. Тоді програма, що здійснює видачу документу, використовує успішно запам'ятовані нею — при створенні або в процесі налаштування користувачем — значення. Таким чином, одна і та ж HTML-сторінка може відображатися по різному на різних машинах, і навіть на одній машині в різні моменти часу.

Зазвичай в HTML-сторінці не обмовляється шрифт текстової частини документу — не вказано ні назва шрифту, ні його розмір. Ясно, що видача на машині з встановленим за замовчуванням значенням розміру шрифту в 10 пунктів буде виглядати зовсім по-іншому, ніж на машині зі шрифтом в 18 пунктів. Навіть при однакових шрифтах у вікнах різного розміру сторінка буде відрізнятися — стане довшою у вузькому вікні.

Другою важливою особливістю мови HTML є можливість включення в документ посилань на інші документи. Зовні посилання виглядає як виділене (наприклад, кольором або підкресленням) місце тексту або малюнка. Поставивши курсор миші на це місце і натиснувши кнопку, користувач отримає на екрані новий документ — той, який був зазначений на посиланні. Таким чином з'являється можливість переходити з одного документу на інший, можливо, навіть розташований на іншій машині.

Більшість елементів мови HTML по своїй суті є дужками. Відкрита дужка визначає початок сторінки, яка має деякі властивості, а закрита — відзначає кінець цієї частини. Все, що знаходиться між дужками є однаковими з точки зору цих властивостей. Наприклад, можна задати для частини тексту колір, що відрізняється від кольору інших частин.

3.1.4 Мова програмування JavaScript

JavaScript (JS) — динамічна, об'єктно-орієнтована мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти [37] з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією [38]. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу [39].

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманий «у спадок» від мови C, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme [40].

3.1.5 Формат передачі даних JSON

В розробленій програмній системі дані між клієнтом і сервером передаються у форматі JSON. Це аргументовано основними перевагами такого формату передачі даних.

Формат JSON — простий і базується на використанні тексту, спосіб зберігати і передавати структуровані дані. За допомогою простого синтаксису розробник має можливість легко зберігати все, що завгодно, починаючи від одного числа до рядків, масивів і об'єктів, в простому тексті. Також можна пов'язувати між собою масиви і об'єкти, створюючи складні структури даних.

Після створення рядка JSON, його легко відправити іншому додатку або в інше місце мережі, так як він являє собою простий текст.

Формат JSON має наступні переваги:

- він компактний;
- його пропозиції легко читаються і складаються як людиною, так і комп'ютером;
- його легко перетворити в структуру даних для більшості мов програмування (числа, рядки, логічні змінні, масиви і так далі);
- багато мови програмування мають функції і бібліотеки для читання і створення структур JSON.

Назва JSON означає JavaScript Object Notation (представлення об'єктів JavaScript). Як і представляє ім'я, він заснований на способі визначення об'єктів (дуже схоже на створення асоціативних масивів в інших мовах) і масивів.

Найчастіше поширене використання JSON — пересилання даних від сервера до браузера. Зазвичай дані JSON доставляються за допомогою AJAX, який дозволяє обмінюватися даними браузеру і серверу без необхідності перезавантажувати сторінку, приклад:

- користувач натискає мініатюру продукту в онлайн магазині;
- JavaScript, що виконується на браузері, генерує запит AJAX до скрипту PHP, запущеного на сервері, передаючи ID обраного продукту;
- скрипт PHP отримує назву продукту, опис, ціну та іншу інформацію з бази даних, а потім складає з даних рядок JSON і відсилає її браузеру;
- JavaScript, що виконується на браузері, отримує рядок JSON, декодує його і виводить інформацію про продукт на сторінці для користувача.

Також можна використовувати JSON для відправки даних від браузера на сервер, передаючи рядок JSON в якості параметра запитів GET або POST. Але даний метод має менше поширення, так як передача даних через запити AJAX може бути спрощена. Наприклад, ID продукту може бути включений в адресу URL як частина запиту GET.

Бібліотека jQuery має кілька методів, наприклад, `getJSON ()` і `parseJSON ()`, які спрощують отримання даних за допомогою JSON через запити AJAX.

Є кілька основних правил для створення рядка JSON:

- Рядок JSON містить або масив значень, або об'єкт (асоціативний масив пар ім'я / значення);
- Масив у квадратних дужках ([i]) і містить розділений комою список значень;
- Об'єкт вставляється в фігурні дужки ({i}) і містить розділений комою список пар ім'я / значення;
- Пара ім'я / значення складається з імені поля, розміщеного в подвійні лапки, за яким слідує двокрапка (:) і значення поля;
- Значення в масиві або об'єкті може бути: числом , логічним значенням (true або false); іншим масивом, інший об'єкт, значення null.

В більшості випадків можна розглядати JSON як альтернативу XML, по крайній мірі, в сфері веб додатків. Концепція AJAX оригінально ґрунтувалася на використанні XML для передачі даних між сервером і браузером. Але в останні роки JSON стає все більш популярним для перенесення даних AJAX [41].

Хоча XML є перевіреною технологією, яка використовується в достатній кількості додатків, перевагами JSON є більш компактний і простий для розпізнавання формат даних.

JSON простий для розуміння і використання, він є досить корисним і гнучким інструментом для передачі даних між додатками і комп'ютерами, особливо при використанні AJAX. Оскільки в розробленій програмній системі було використано технологію AJAX, відповідно для кращої взаємодії цієї технології було використано саме формат даних JSON.

3.1.6 Технологія комунікації сервера та клієнта AJAX

Процес отримання деякої сторінки у всесвітній павутині завжди починається з того, що користувач вводить в адресному рядку браузера URL або переходить за посиланням, що містить URL. Браузер, отримавши цей URL, відправляє запит на веб-сервер. Веб-сервер отримавши цей запит, формує клієнту (браузеру) відповідь і відправляє його. Браузер, отримавши відповідь (веб-сторінку) з веб-сервера, відображає його в своєму вікні (вкладці).

Без технології AJAX, єдиним рішенням в цьому випадку було б відправка запиту на сервер і отримання точно такої же сторінки, але з іншими слайдами зображень.

Цей варіант виглядає непрактично, але цей був єдиний спосіб до появи технології AJAX [42]. Тобто замість того щоб оновити тільки потрібний блок інформації на сторінці, ми запитуємо натомість нову сторінку у сервера. Це призводить до того, що користувач чекає, поки завантажиться сторінка з новими зображеннями, що видається не дуже зручним.

На рисунку 3.3 зображено порівняння класичної моделі і моделі з використанням AJAX.

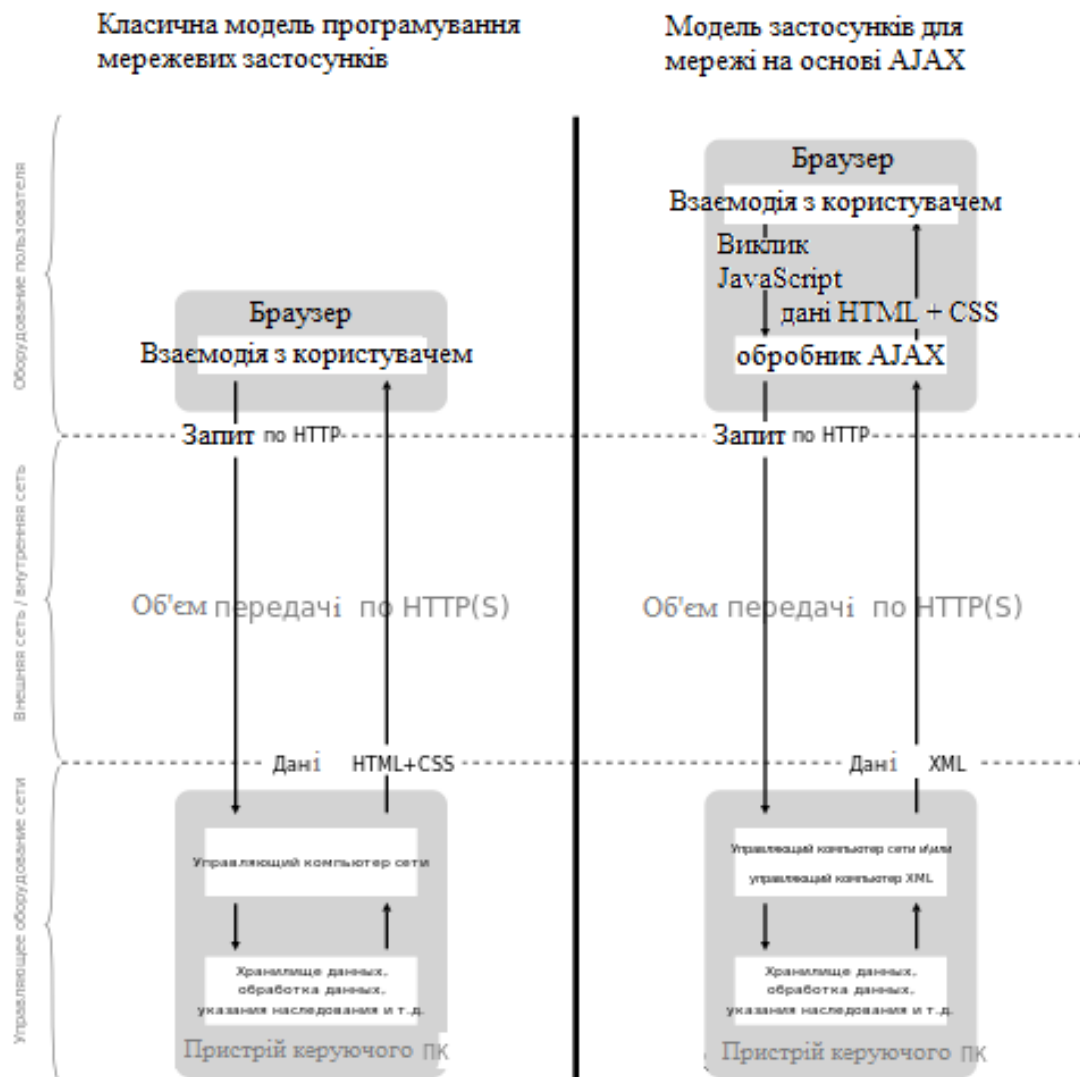


Рисунок 3.3 — Порівняння AJAX та класичної моделі

Застосування технології AJAX на сторінці дозволить користуватися такими перевагами:

- зменшити трафік;
- знизити навантаження на сервер;
- прискорити чуйність інтерфейсу сторінки і його мерехтіння;
- поліпшити інтерактивність (тобто більш зручно вести діалог (здійснювати взаємодію) з користувачем).

3.1.7 Технологія Razor

При використанні елементів коду C # в уявленнях ми використовували знак @. Це знак, який використовується двигком уявлень Razor для переходу до коду на мові C # / VB.NET.

При виклику методу View контролер не виробляє рендеринг уявлення і не генерує розмітку html. Контролер тільки готує дані і вибирає, яке уявлення треба повернути в якості об'єкта ViewResult. Потім вже об'єкт ViewResult звертається до двигку уявлення для рендеринга уявлення в вихідний результат.

До MVC 3 використовувався тільки двигок Web Forms, який використовував стандартний синтаксис ASP.NET. наприклад:

```
<% foreach(BookStore.Models.Book b in ViewBag.Books) { %>
  <li>Книга: <%= b.Name %></li>
<% } %>
```

Введення в MVC 3 двигка Razor дозволило зменшити синтаксис при виклику коду C #, зробити код більш "чистим". Аналогічний приклад на Razor виглядатиме так: @foreach (BookStore.Models.Book b in ViewBag.Books) { Книга: b.Name </ li>} Тому його використання більш переважно. В кінцевому рахунку, вибір двигка уявлень залежить від особистих переваг, і обидва двигки по функціональності рівноцінні.

В даному випадку важливо розуміти, що двигок уявлень, наприклад, Razor - це не якийсь новий мову, це лише спосіб рендеринга уявлень, який має певний синтаксис для переходу від розмітки html до коду C #.

Крім движка Razor можна використовувати і движки сторонніх виробників. Наприклад, движок Spark, який використовує декларативний синтаксис, або Brail, який використовує мову Boo.

Основи синтаксису Razor

Використання синтаксису Razor характеризується тим, що перед виразом коду стоїть знак @, після якого здійснюється перехід до коду C#. Існують два типи переходів: до виразів коду і до блоку коду.

Наприклад, перехід до вираження коду:

```
<p>@b.Name</p>
```

azor автоматично розпізнає, що Name - це властивість об'єкта b.

Також можна використовувати стандартні класи і методи, наприклад, виведемо поточний час:

```
<H3> @ DateTime.Now.ToShortTimeString () </ h3>
```

Застосування блоків коду аналогічно, тільки знак @ ставиться перед усім блоком коду, а движок автоматично визначає, де цей блок коду закінчується:

```
@foreach (BookStore.Models.Book b in Model)
```

```
{ <P> @ b.Name </ p>}
```

Більш того ми можемо створювати блоки коду в поданні, створювати там змінні так само, як і в файлі коду C#:

```
string head = "Привіт світ !!!";
```

```
head = head + "Ласкаво просимо на сайт!";
```

```
}
```

```
<H3> @head </ h3>
```

Для створення однакового виду сайту застосовуються майстер-сторінки. Майстер-сторінки - це по суті ті ж самі уявлення. Ми можемо визначити на майстер-сторінці деякі елементи, які будуть відображатися на всіх сторінках сайту. А також ми можемо визначити наповнювачі або плейсхолдери, зміст яких забезпечують інші уявлення.

3.1.8. Середовище розробки MS Visual Studio 2017

MS Visual Studio 2017 — комерційне інтегроване середовище розробки для різних мов програмування від компанії Microsoft. Система поставляється у вигляді урізаної по функціональності безкоштовної версії “Community Edition” і повнофункціональної комерційної версії “Ultimate Edition”, для якої активні розробники відкритих проєктів мають можливість отримати безкоштовну ліцензію.

До складу Community версії входить модуль візуального проєктування GUI-інтерфейсу Swing UI Designer, XML-редактор, редактор регулярних виразів, система перевірки коректності коду, система контролю за виконанням завдань і доповнення для імпорту та експорту проєктів з Eclipse. Доступні засоби інтеграції з системами відстеження помилок JIRA, Trac, Redmine, Pivotal Tracker, GitHub, YouTrack, Lighthouse.

Під час розробки даного програмного забезпечення було використано 2017.1 версію продукту.

Її особливості:

— відладчик:

— вирази Groovy в Evaluate Expression і Watches тепер можна використовувати при налагодженні Java коду;

— більш зручне налагодження декількох потоків;

— попередження при розбіжності вихідного коду із запущеною копією;

— системи контролю версій;

— підтримка Git worktrees;

— більш зручний мердж і порівняння, завдяки підсвічуванню змін на рівні слів;

— редактор:

— зрушення виразів вправо і вліво;

— автоматичний імпорт статичних методів і констант;

— підтримка мов, що записуються справа наліво (наприклад, арабської та іврити);

— технологія Gradle:

— підтримка custom source sets: тепер кожен source set представлений окремим модулем і може мати власні залежності (в результаті було виправлено величезну кількість проблем);

— автоматична конфігурація EAR-артефактів;

3.1.9. Система управління базами даних Microsoft SQL Server Express

Microsoft SQL Server Express є версією системи керування реляційною базою даних Microsoft SQL Server, яку можна безкоштовно завантажувати, розповсюджувати та використовувати. Він містить базу даних, спеціально призначену для вбудованих і менш масштабних додатків. Продукт простежує свої корені в продукті Microsoft Database Engine (MSDE), який був доставлений в SQL Server 2000. Брендінг "Експрес" використовувався з моменту випуску SQL Server 2005.

SQL Server Express надає безліч функцій платних повних версій системи керування базами даних Microsoft SQL Server. [43] Однак він має технічні обмеження, що робить його непридатним для деяких масштабних розгортань. Відмінності в експрес-товарі включають:

Максимальний розмір бази даних - 10 Гб на базу даних в SQL Server 2016, SQL Server 2014, SQL Server 2012 та 2008 R2 Express [44] (4 Гб для SQL Server 2008 Express та раніше, у порівнянні з 2 Гб у колишньому MSDE). Ліміт застосовується до бази даних (файли журналів виключаються); але в деяких сценаріях користувачі можуть отримати доступ до більшої кількості даних через використання декількох взаємопов'язаних баз даних.

Немає служби агента SQL Server.

Обмеження використання штучного обладнання:

— Одиночний фізичний процесор, але декілька ядер дозволено ;

— 1 Гб оперативної пам'яті (працює на системі з більшою кількістю оперативної пам'яті, але використовує лише не більше 1 Гб на екземпляр SQL Server Database Engine. "Рекомендовано: Експрес-видання: 1 Гб Всі інші видання: принаймні 4 Гб [45], розмір бази даних збільшується, щоб забезпечити оптимальну

продуктивність "). Експрес із розширеними службами обмежує 4 Гб на екземпляр служб звітування (не доступний для інших варіантів Express). Служби аналізу не доступні для будь-якого експрес-варіанта.

На відміну від попереднього продукту MSDE, експрес-продукт не включає паралельного робочого навантаження-губернатора, щоб "обмежити продуктивність, якщо двигун бази даних отримує більше роботи, ніж це характерно для невеликої кількості користувачів" [46].

SQL Server Express включає в себе кілька GUI-інструментів для управління базами даних. До них відносяться:

- SQL Server Management Studio - починаючи з 2012 р. SP1; перед цим надана лише вилучена версія SQL Server Management Studio Express;

- SQL Server Configuration Manager;

- Інструмент конфігурації області майданчика SQL Server;

- SQL Server Business Intelligence Development Studio;

- У попереднього продукту MSDE, як правило, бракує базових інструментів керування графічним інтерфейсом .

- Функції, доступні в SQL Server "Standard" та кращі видання, але відсутні в SQL Server Express включають в себе:

- Служби аналізу;

- Інтеграційні послуги;

- Служби сповіщення.

3.2. Опис бази даних

База даних включає в себе опис двох великих компонентів. Перший це опис вакансії і другий це опис профілю вузу.

Також база даних має таблицю, яка зберігає інформацію про користувачів системи. Оскільки в системі є можливість реєструватися в двох режимах:

1. Роботодавець

2. Представник вузу

В базі даних зберігаються вакансії, які створюють роботодавці.

База даних складається з чотирьох основних таблиць: таблиці «Вакансія», «Реєстрація», «Навики для вакансій» та «Інформація про користувача», «Вуз», «Спеціалізація», «Навики в вузі». Структура бази даних та таблиць зображена на рисунку 3.4.

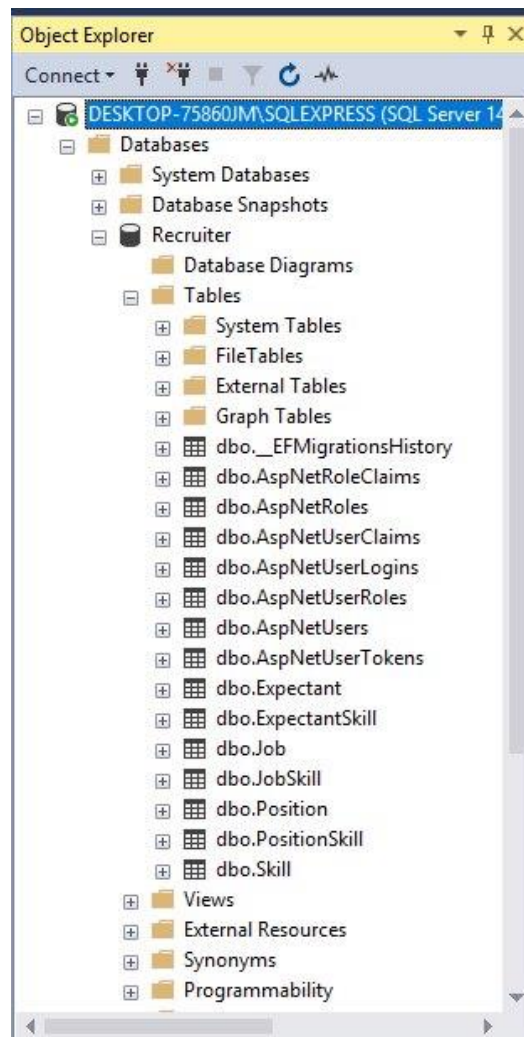


Рисунок 3.4. — Структура бази даних та таблиць системи корегування навчальних планів в залежності від потреб ринку інформаційних технологій

Однією з важливих таблиць бази даних, є інформація про користувачів системи (рисунок 3.5). Варто зазначити, що на поле “e-mail” накладається обмеження по кількості символів та типу введення даних за певним шаблоном. Кожен користувач має свій тип, які записані перелічуваним типом 1 або 2, рекрутер

та представник вузу відповідно.

1	id_iser	name	surname	last_name	E-mail	phone_number	password	user_type
2	1	Хохлова	Яніна	Геннадіївна	yanina.Khokhlovs@gmail.com	637834452	1D@dEVOX	1
3	2	Біченко	Дмитро	Леонідович	Buchenko@gmail.com	786459321	1D@dEVOX	2
4	3	Масечко	Ірина	Олександрівна	muslra@gmail.com	9678541236	1D@dEVOX	1
5	4	Байда	Дмитро	Андрійович	Baida@gmail.com	934526389	1D@dEVOX	1
6	5	Водзянський	Олександр	Олегович	Oleeksa@gmail.com	637823697	1D@dEVOX	1
7	6	Лучков	Владислав	Феликсович	FeliksVlad@gmail.com	78965412	1D@dEVOX	2
8	7	Хохлова	Геннадій	Анатолійович	Genho@gmail.com	1236578	1D@dEVOX	2
9	8	Пустовий	Богдан	Андрійович	bodyan95@gmail.com	78965412	1D@dEVOX	2

Рисунок 3.5 — Приклад заповнення таблиці «Користувач»

На рисунку 3.6. зображено приклад заповнення таблиці вакансії. Варто зауважити, що в цій таблиці за кожною вакансією закріплені навички, які необхідно мати, щоб отримати цю позицію.

1	VacancyID	VacancyName	Company	ProjectInfo	Skills
2	1	Junior Front End Developer	EPAM	SimCorp offers an interesting and challenging position in an international, dynamic and professional organization. You will be working in an organization that offers great flexibility with regards to planning your own work and great development opportunities professionally and personally.	1
3	2	Junior Java Developer	EPAM	People.ai helps sales, marketing and customer success teams uncover every revenue opportunity from every customer. Companies like New Relic, Tanium, Lyft, Okta, and Mulesoft choose People.ai to capture all customer contacts, activity and engagement to drive actionable insights across all Enterprise revenue.	2
4	3	Junior PHP Developer	Luxsoft	InfoSuite A/S (www.infosuite.com) develops, sells and implements Business Intelligence and Budgeting solutions for medium-sized companies in Denmark as well as internationally. InfoSuite headquarter is placed in Denmark. E	3
5	4	Junior C# Developer	SoftServe	Ability to work independently with minimal supervision and collaborate effectively with other team members as-needed Excellent planning, organizational, and time management skills with high attention to detail in a demanding, deadline-driven environment Drive, passion, and initiative.	4
6	5	Junior QA	QubitLabs	Recently recognized as a 2016 Technology Pioneer by the World Economic Forum, Upskill is the industry leader in wearable technology for the connected workforce. Our award-winning software platform, Skylight, powers wearable devices such as smart glasses, and is used by some of the largest and most recognized companies in the world.	5
7					

Рисунок 3.6 — Приклад заповнення таблиці «Вакансія»

На рисунку 3.7 зображений приклад заповнення таблиці з навиками. Ця таблиця формується автоматично після того, як користувач додає новий, раніше невідомий навик. Ці навички в подальшому використовуються для формування статистики.

1	SkillID	SkillName
2	1	Java
3	2	C#
4	3	Spring
5	4	Hibernate
6	5	MAven
7	6	PHP
8	7	MySQL
9	8	OOP

Рисунок 3.7 — Приклад заповнення таблиці «Навик»

В системі передбачене задання оцінки важливості скілів. Необхідно було передбачити зберігання цих оцінок. Для цього було створено окрему таблицю

(рисеуїг 3.8) в якій зберігається інформація про належність навичку до певної вакансії та яку оцінку виставив користувач цьому навичку.

1	skill_id	vacancy_id	points
2	1	4	5
3	1	5	3
4	1	5	2
5	1	5	1
6	5	5	1
7	4	7	4
8	7	8	2
9	8	1	3

Рисунок 3.8 — Приклад заповнення таблиці «Дані»

3.3 висновки до розділу 3

В розділі номер три були розглянуті основні інструменти для розробки системи корегування навчально плану в залежності від потреб ринку інформаційних технологій.

При обранні мови для Back End розробки були враховані найважливіші фактори для розробки великого веб сервісу. Врахували можливості зниження навантаження на сервер, легка конфігурація використання в хмарі, кроссплатформенність, потужна система маршрутизації.

При обранні технологій для Front-End розробки були враховані такі фактори, як легкість у створенні веб інтерфейсу, легке корегування, зручна інтеграція з мовою Back End розробки.

Для виконання дипломної роботи важливо провести аналіз існуючих інструментів для виконання поставлених задач. Це буде впливати на час виконання, якість, швидкість роботи продукту та її якість.

При розробці і реалізації програмного продукту я використала інтегровану систему розробки Visual Studio. В якості технології Back End розробки - .NET Core, Front-End – Razor, CSS, HTML.

Було представлено технології, завдяки яким було реалізовано систему корегування навчального плану в залежності від потреб ринку інформаційних технологій

4. ВЗАЄМОДІЯ З КОРИСТУВАЧЕМ

У розділі 4 будуть розглянуті основні можливості взаємодії користувача з системою. Будуть розглянуті системні вимоги та інструкція по інсталяції програмної системи. У випадку останнього, користувачеві не потрібно встановлювати ніяких додаткових інструментів, оскільки сервіс являє собою веб-ресурс у вільному доступі. Для того, щоб можна було користуватись системою корегування навчального плану в залежності від потреб ринку потрібно встановити додаткові інструменти.

4.1. Інсталяція та налаштування програмного продукту

Для нормальної роботи програмного продукту необхідне виконання мінімальних системних вимог:

1. IBM-сумісний комп'ютер з процесором класу Intel Core та вище й тактовою частотою 1.5 Гц.
2. Об'єм оперативної пам'яті не менше 1Гб.
3. Доступ до мережі Інтернет зі швидкістю не менше 1Мбіт/сек.
4. 2Гб дискового простору

Для встановлення системи, потрібна наявність наступних системних засобів:

1. Операційна система MS Windows (XP, 7, 8, 8.1, 10), чи дистрибутив Linux (Ubuntu, Debian, BSD), Mac OS.
2. Сервер MS SQLServer.
3. Встановлений пакет .NET 4.0.

У разі потреби використання стосунку для розробки, при потребі у виправленні недоліків чи покращення функціоналу, потрібно встановити усі необхідні пакети для нормального функціонування застосунку. Це можна зробити з використанням команди `npm install`. Також в якості альтернативного інструменту

для роботи з Javascript пакетами можна використовувати утиліту npm.

4.2. Опис програмного продукту

Основною функцією стало аналіз IT ринку і порівняння існуючих університетських програм.

Програмний продукт збирає інформацію по Job - сайтах і робить аналіз, які вакансії зараз популярні на ринку.

Спочатку всі вакансії розділяються по основним мовам програмування:

Після цього розподілення в кожній групі починають оброблятися вакансії і ті технології зустрічається необхідність та знання, які найчастіше зустрічаються з тією чи іншою мовою.

Наприклад для Java, роботодавці найчастіше хочуть бачити знання Spring Framework, Maven, Hibernate, об'єкно орієнтоване програмування, MySQL, IntelegeIDEA.

Такий аналіз ринку варто робити 1 раз на 2-3 місяці так як це обробка достатньо великого об'єму даних.

Після того як у нас є оброблені данні по потребам ринку інформаційних технологій.

То в програмі забезпечена функція виводу статистики по необхідності знань для того, що б претендувати на посаду.

Отже програмний продукт створений для порівняльного аналізу важливих навиків для того, що б влаштуватися на роботу і навчальних програм в університетах.

Відбувається парсинг по навикам, які здобуває студент після закінчення навчального закладу. По черзі знаходить ключові слова (слова, які найчастіше зустрічаються у вакансіях) і заносяться оцінки для кожної вакансії. За кожне співпадіння надається 1 бал. В кінці ми отримає найбільш підходящі вакансії для вузів і на скільки вона підходить у відсотках. Також показується, які навики необхідно додати, що б отримати спеціаліста, який зможе працювати на данній

позиції.

4.3. Сценарій роботи користувача з програмою

Отже далі можна побачити сценарій роботи користувача з системою. Оскільки це веб-додаток, то людина має зареєструватися у системі або зробити вхід в систему. Рисунок 4.1 ілюструє стартову сторінку користувача. На ній можна побачити кнопки входу в систему та перегляду основної сторінки.

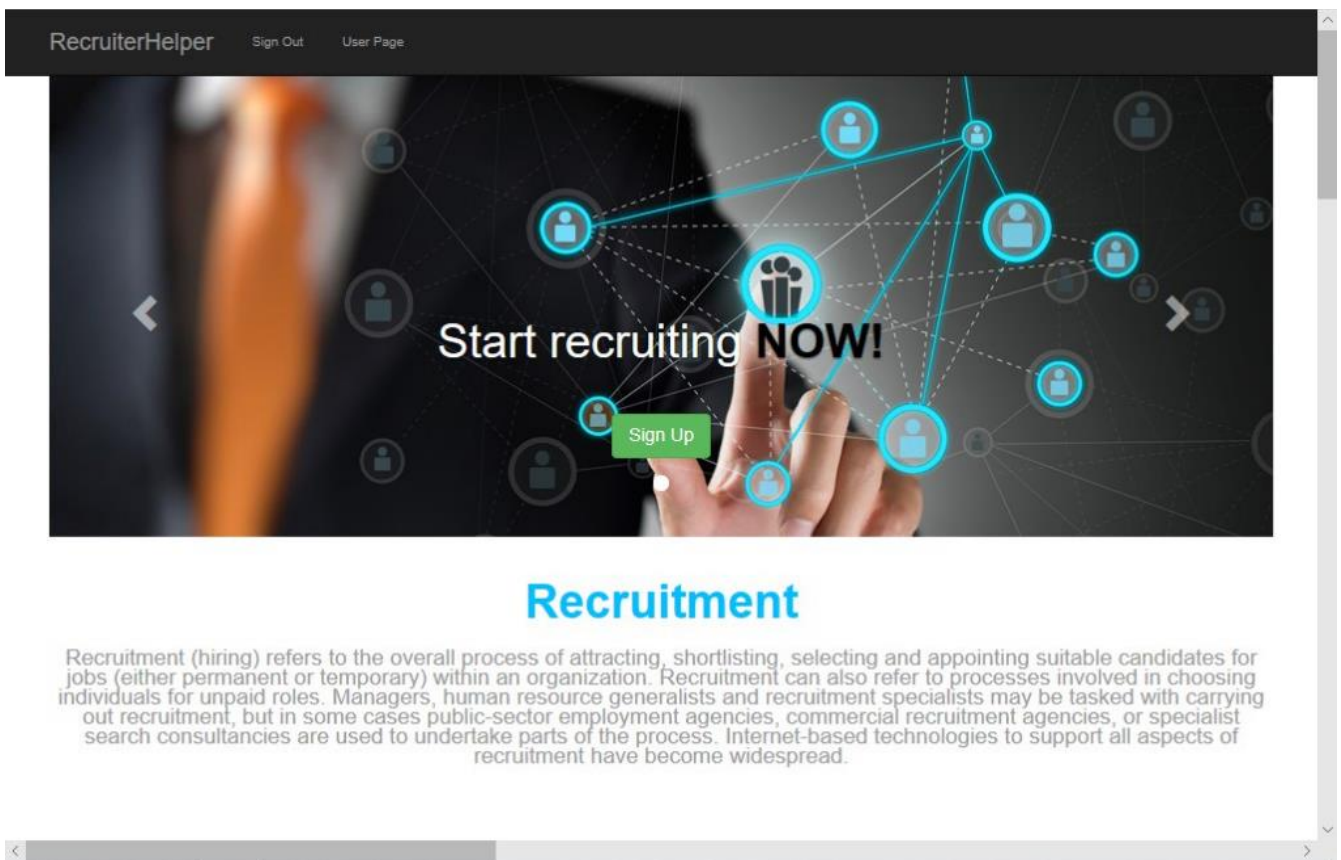


Рисунок 4.1 — Стартова сторінка користувача

Після натискання кнопки зареєструватися, користувач попадає на екран, де необхідно ввести свої дані для реєстрації (рисунок 4.2) . Дані перевіряються на коректність. Користувачу пропонується дружній інтерфейс завдяки якому, користувач легко розуміє, що йому необхідно натиснути та які дані ввести. Система допомагає підказками у впливаючих вікнах.

Система надає підказки, чи вірно введені дані, у випадку помилок, користувач може побачити червоний колір, який вкаже на місце помилки. Демонстрація невірно введених даних зображено на рисунку 4.3.

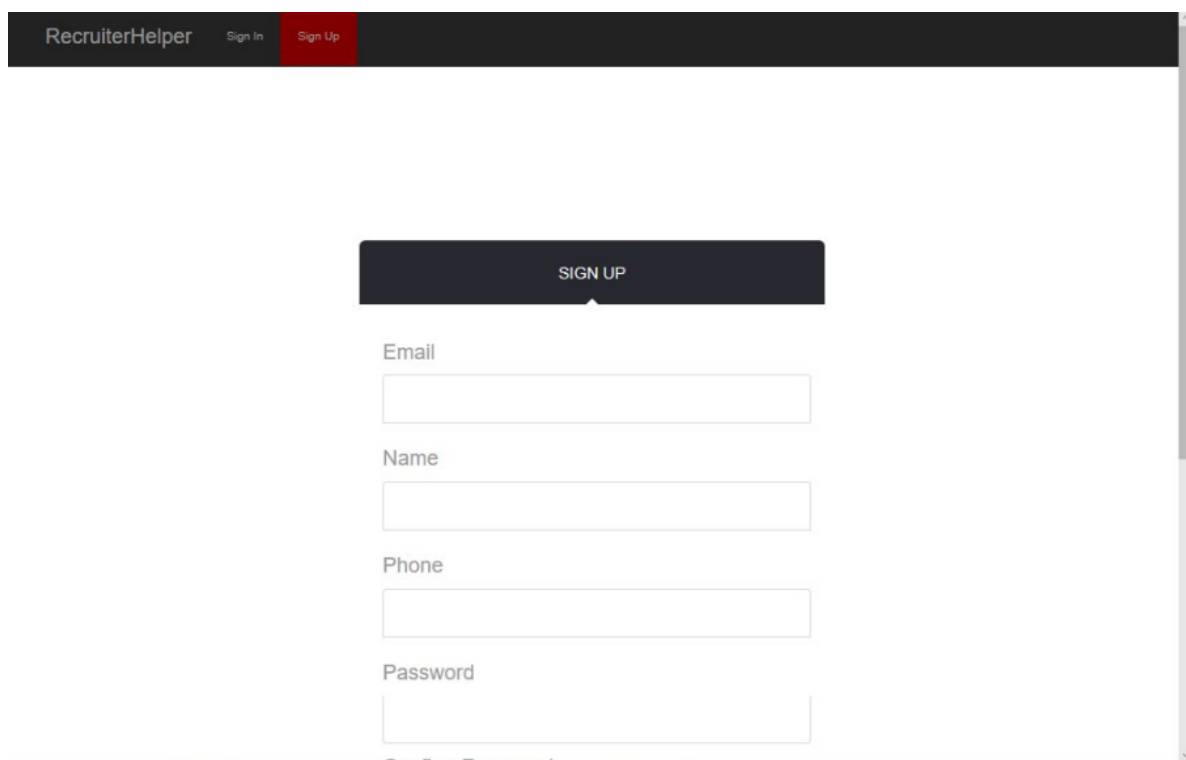
A screenshot of a web application's registration form. At the top, there is a dark navigation bar with the text 'RecruiterHelper' and two buttons: 'Sign In' and 'Sign Up'. The 'Sign Up' button is highlighted in red. Below the navigation bar is a dark button labeled 'SIGN UP'. Underneath, there are four input fields labeled 'Email', 'Name', 'Phone', and 'Password'. The 'Confirm Password' label is partially visible at the bottom of the form.

Рисунок 4.2 — Форма реєстрації

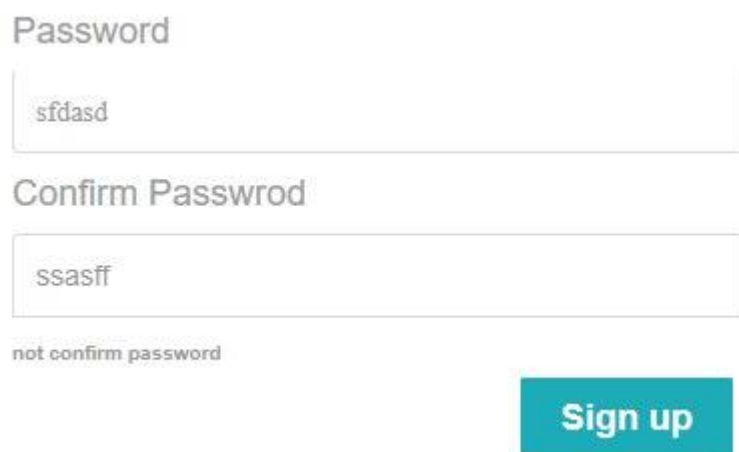
A close-up screenshot of the registration form focusing on the password fields. The 'Password' field contains the text 'sfdasd' and the 'Confirm Password' field contains 'ssasff'. Below the 'Confirm Password' field, there is a red error message that reads 'not confirm password'. A teal 'Sign up' button is positioned to the right of the error message.

Рисунок 4.3— Вивід некоректних даних

Якщо користувач вже зареєстрований в системі, то він може скористатися формою входу, яка представлена на рисунку 4.4. В ній користувач має ввести логін

та пароль. У випадку невірної вводу користувач побачить помилку. Також передбачена кнопка «Забули пароль». Скориставшись нею, користувач отримає лист на пошту з новим паролем і буде мати можливість поновити його.

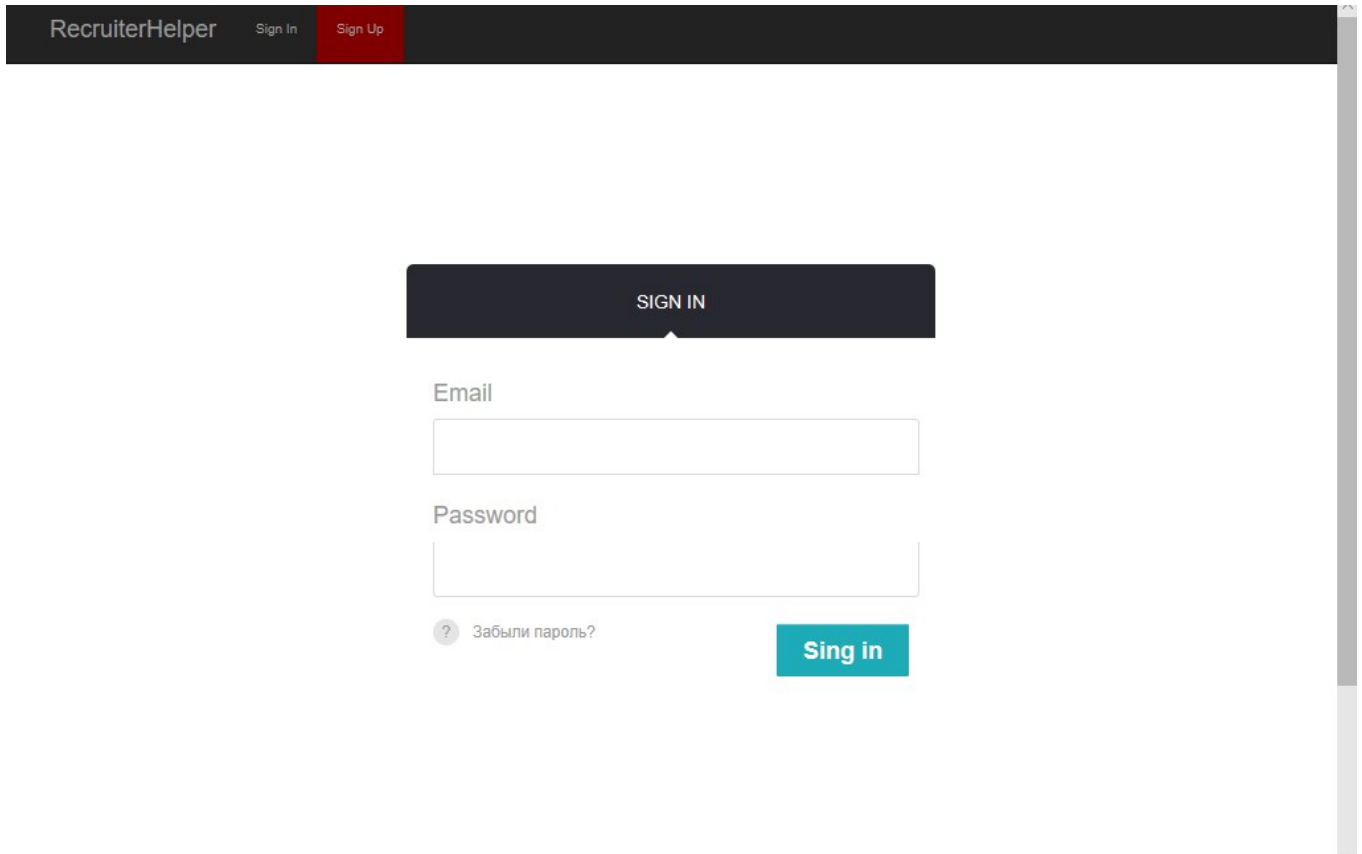
The image shows a screenshot of a web application's login interface. At the top, there is a dark navigation bar with the text 'RecruiterHelper' on the left, and two buttons: 'Sign In' and 'Sign Up'. The 'Sign Up' button is highlighted in red. Below the navigation bar, there is a dark grey button labeled 'SIGN IN'. Underneath this button are two input fields: 'Email' and 'Password'. Below the 'Password' field, there is a link that says '? Забули пароль?' (Forgot password?). To the right of this link is a teal button labeled 'Sing in'.

Рисунок 4.4 — Форма входу в систему

Увійшовши в систему, користувач бачить головну сторінку з інформацією про себе (рисунок 4.5). Також він може перейти на головну сторінку, внести зміни до свого профілю та скористатися вкладкою «Вакансія», яка знаходиться у верхній частині екрану.

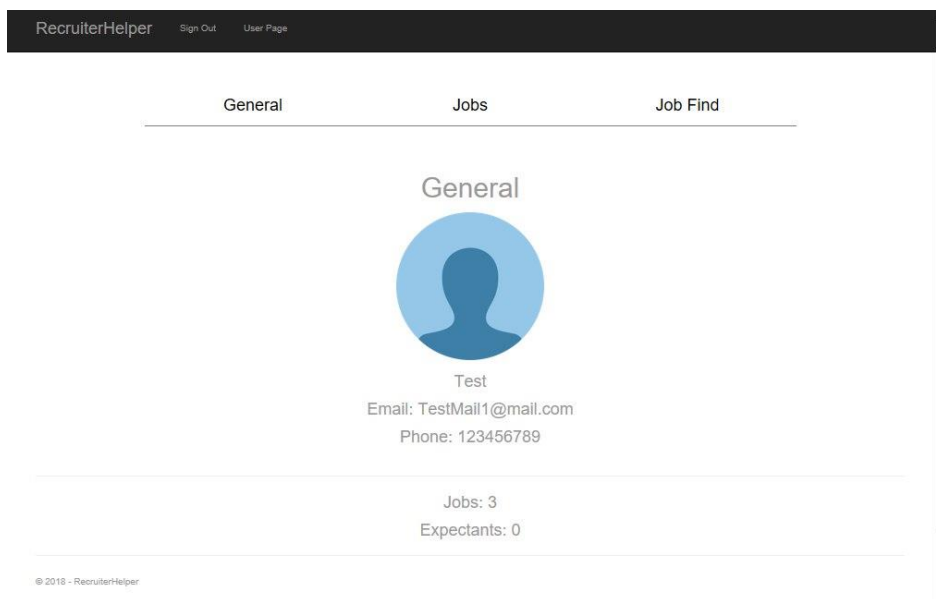


Рисунок 4.5 — Сторінка профілю користувача

Натиснувши кнопку «Jobs» користувач потрапляє на вікно де може проглянути які є вакансії, додати вакансію, видалити вакансію та редагувати її (рисунок. 4.6).

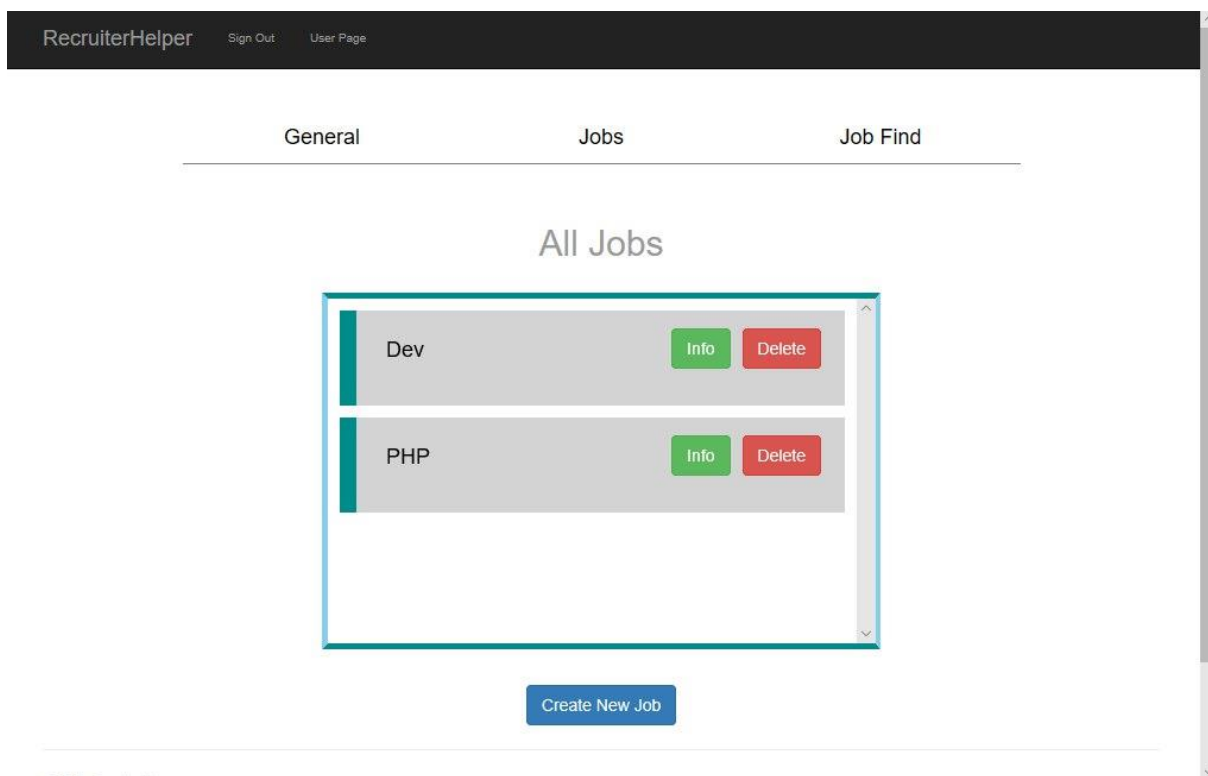


Рисунок 4.6 — Вікно управління вакансіями

Якщо натиснути на кнопку «Створити нову вакансію», користувач системи, потрапляє у вікно, де може описати вакансію, задати навички які необхідні та виставити їм оцінки (рисунок 4.7).

RecruiterHelper Sign Out User Page

General Jobs Job Find

Create Vacancy

Job name:

Project description:

Skills:

Name	Points of importance (0-5)
<input type="text" value="Java"/>	<input type="text" value="5"/>
<input type="text" value="Spring"/>	<input type="text" value="3"/>
<input type="text" value="Hibernate"/>	<input type="text" value="3"/>



Qubit Labs

Рисунок 4.7 — Вікно створення нової вакансії


Натиснувши кнопку “Job Find” (рисунок 4.8) можна завантажити своє CV та натиснувши кнопку “Analyze” користувач побачить розпарсене резюме і вакансію, яка найбільше підходить його профілю (рисунок 4.9).

RecruiterHelper Sign Out User Page

General Jobs Job Find

Job Find

1. Upload CV file in supported format.
2. Analyze.



Choose File

Рисунок 4.8 — Сторінка пошуку роботи

Рисунок 4.9 — Форма завантаження резюме

В залежно від степені співпадіння, користувач може побачити відсоткову приналежність до вакансії. І програма підкаже яка вакансія вам найбільше підходить. Є три градації підходження на вакансії 0%-30% - погано підходить, 30%-60% - Нормально підходить, 60%-100% - дуже гарно підходить (рисунки 4.10 – 4.13).

Рисунок 4.10 — Результати парсингу резюме

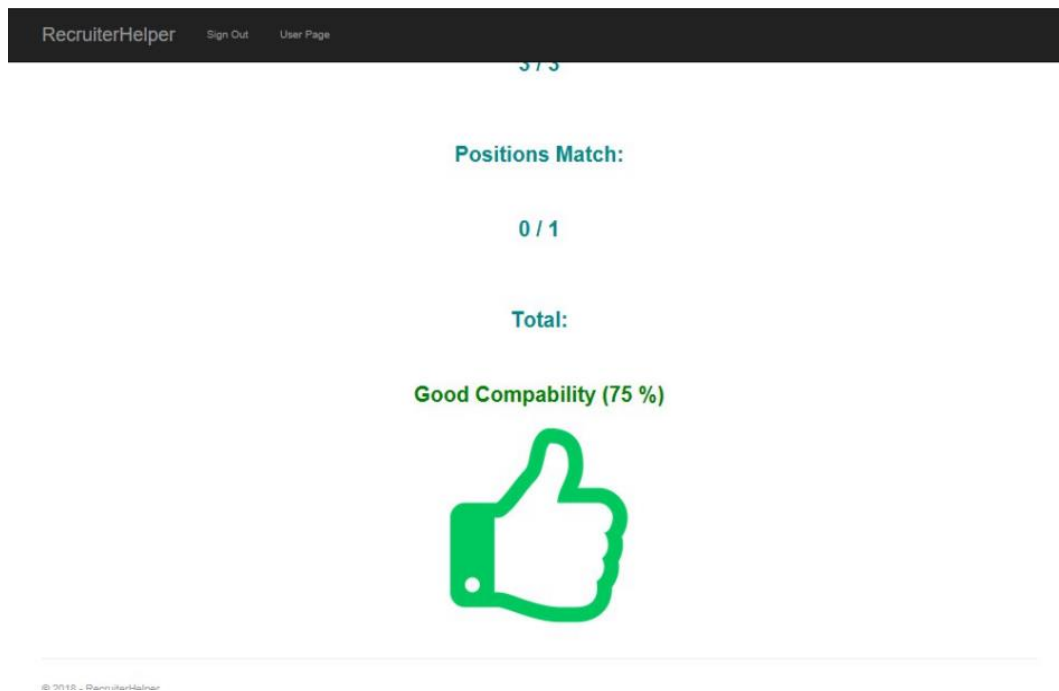


Рисунок 4.11 — Результаты парсингу резюме при позитивному результаті

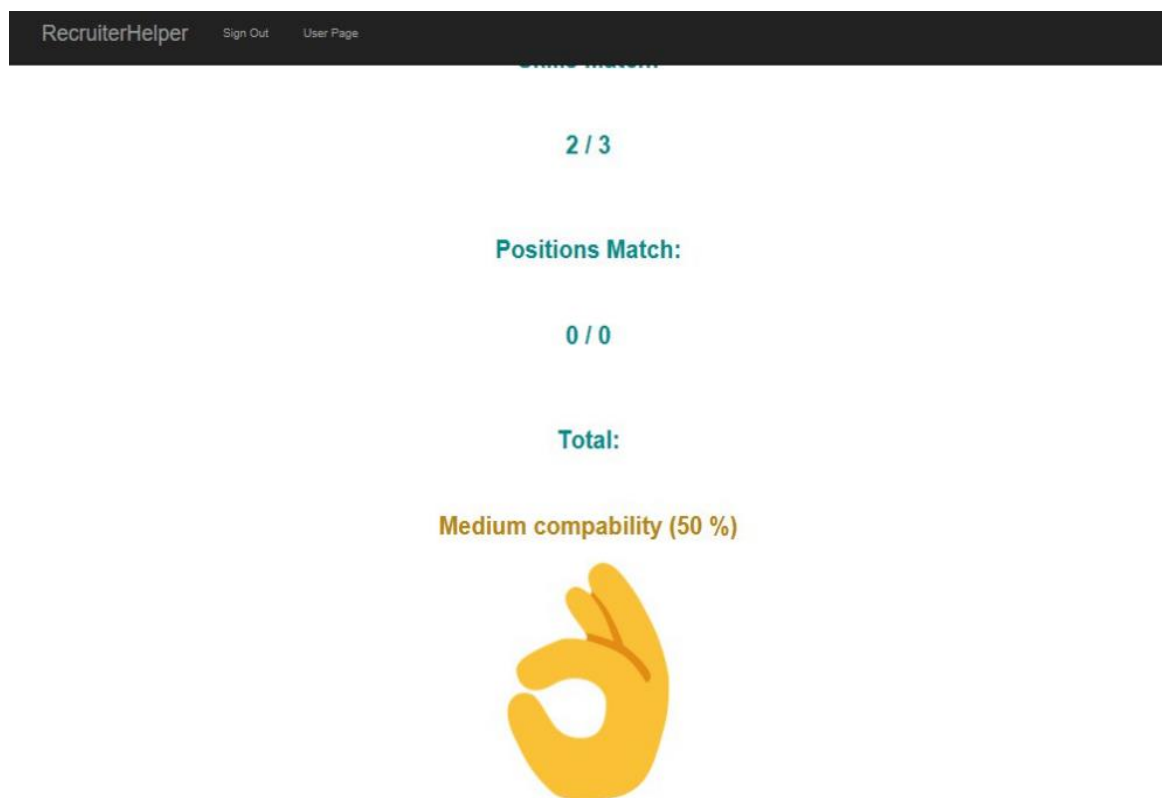


Рисунок 4.12 — Результаты парсингу резюме при нейтральному результаті

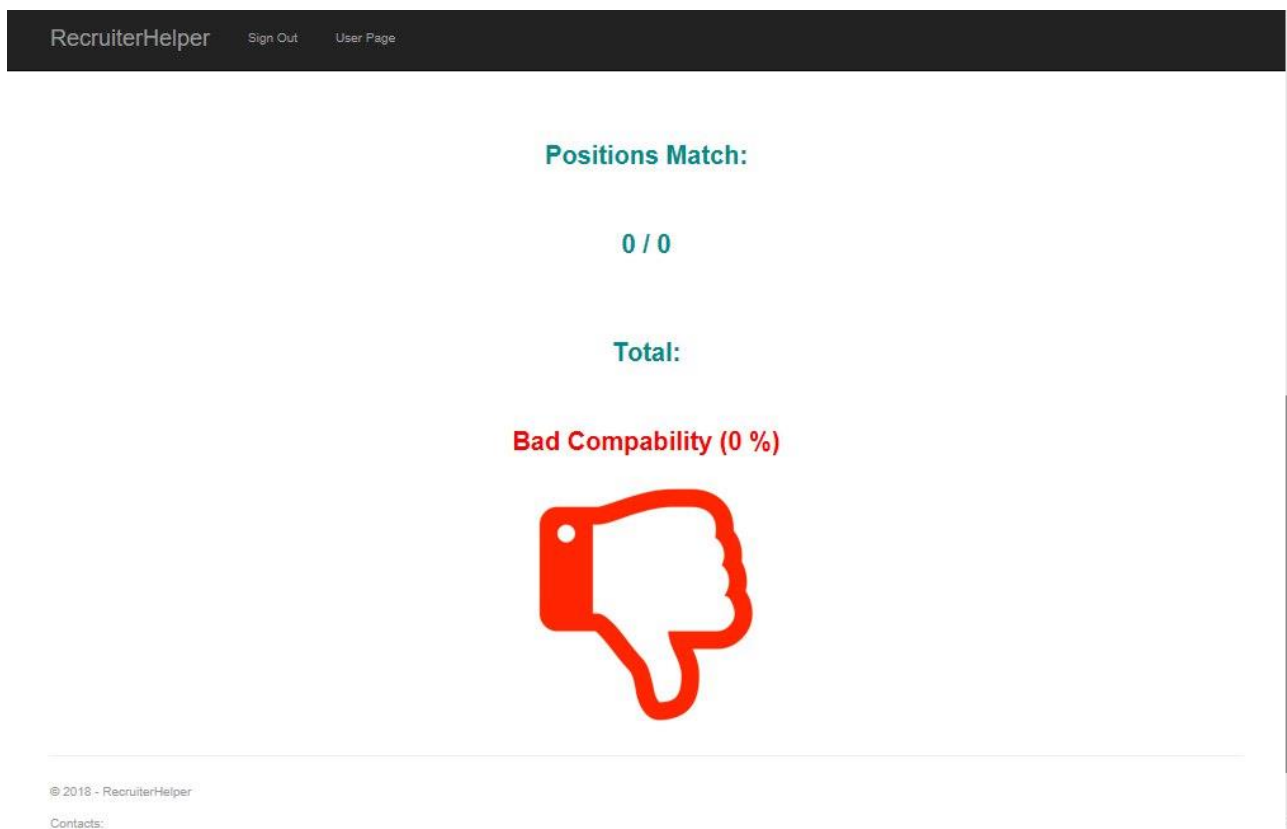


Рисунок 4.13 — Результати парсингу резюме при негативному результаті

Отже система рекомендує на яку вакансію найбільше підходить випускник вузу. А також повідомляє представника вузу, які навички необхідно додати, що б студент підходив на вакансію на 100%.

4.4. Висновки до розділу 4

У розділі номер 4 представлена покрокова інструкції щодо використання системи корегування навчального плану в залежності від потреб ринку інформаційних технологій.

Показано можливі варіанти взаємодії користувача з системою, а саме реєстрація користувача, додавання нової вакансії, перегляд існуючих вакансій, результати порівняння навичок випускника вузу та навичок необхідних для заняття відкритої вакансії.

5. СТАРТАП ПРОЕКТ

Розділ має на меті проведення маркетингового аналізу стартап проекту для визначення принципової можливості його ринкового впровадження та можливих напрямів реалізації цього впровадження. Проведення маркетингового аналізу передбачає виконання нижченаведених кроків.

5.3 Опис ідеї проекту

В межах підпункту слід проаналізувати та подати у вигляді таблиць:

1. Зміст ідеї (що пропонується).
2. Можливі напрямки застосування.
3. Основні вигоди, що може отримати користувач товару.
4. Чим відрізняється від існуючих аналогів та замінників.

Перші три пункти подаються у вигляді таблиці (таблиця 5.1) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки.

Таблиця 5.1. — Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Сегменти споживачів
Формування сценаріїв розвитку електроенергетики методами форсайту	1.Визначення потреб ринку інформаційних технологій	ВНЗ
	2.Забезпечення можливості компаніям у виборі ВНЗ для вигідної співпраці	Компанії в сфері інформаційних технологій

Таблиця 5.1.(продовження) — Опис ідеї стартап-проекту

	3.Системи, які використовують методи форсайту	ВНЗ
--	---	-----

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

1. Визначення переліку техніко-економічних властивостей та характеристик ідеї.

2. Визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку [47].

3. Проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (таблиця 5.2).

Таблиця 5.2. — Визначення сильних, слабких та нейтральних характеристик

№ п/п		(потенційні) товари/концепції конкурентів			
		Мій проект	CRM системи	Системи рекрутинга	Системи
1	W слабка сторона	Відносно повільний час роботи	Повільний час роботи	Не є направленими на конкретну предметну область.	Не враховує особливості потреб певних груп людей

Таблиця 5.2 (продовження) — Визначення сильних, слабких та нейтральних характеристик

2		Відсутність обробки наявних вакансій з мережі інтернет	Матеріали додають лише зареєстровані користувачі	Не враховує особливості формування сценаріїв	Обрахунки не автоматизовані
3	N нейтральна сторона	Відсутність можливості робити незареєстрованих користувачів	Надто багато функцій	Можливість задавати валідацію даних	Проводиться спеціально нанятим персоналом
4	N нейтральна сторона	Невелика ресурсозатратність на стороні користувача	Експерт може користуватися лише готовою схемою заповнення вакансій	Експерт може користуватися лише готовою схемою заповнення вакансій	Експерт може користуватися лише готовою схемою заповнення вакансій
5	S сильна сторона	Кроссплатформний застосунок	Формування сценарію відбувається автоматично	Можливо використовувати в різних предметних областях	Формування більш детального сценарію

Таблиця 5.2 (продовження) — Визначення сильних, слабких та нейтральних характеристик

	Дуже швидке оновлення технологій забезпечує легке впровадження нових функцій	Побудований сценарій можна представити у вигляді діаграм для більшої наочності	Присутні шаблони заповнення вакансій та профілів кандидатів	
	Існує можливість порівняльного аналізу навиків у випускника та потреб ринку	Існує можливість автоматизувати деякі дії		

5.2 Технологічний аудит ідеї проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту. Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 5.6):

1. За якою технологією буде виготовлено товар згідно ідеї проекту.
2. Чи існують такі технології, чи їх потрібно розробити/доробити.
3. Чи доступні такі технології авторам проекту.

Таблиця 5.3. — Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології і реалізації	Наявність технологій	Доступність технологій
1	Інтерфейс користувача	Мова програмування C#	Наявна	Умовна безкоштовно

Таблиця 5.3. (продовження)—Технологічна здійсненність ідеї проекту

2	Розширювана база даних	SQL, MS Server	Наявна	Умовна безкоштовно
3	Алгоритм створення анкет	Мова програмування С#	Відсутня	Відсутня
4	Алгоритм формування сценарію	Мова програмування С#	Відсутня	Відсутня
<p>Висновок: проект реалізувати можливо. Обрана технологія реалізації ідеї проекту: Формування сценаріїв розвитку електроенергетики методами форсайт</p>				

За результатами аналізу таблиці робиться висновок щодо можливості технологічної реалізації проекту: так чи ні, а також технологічного шляху, яким це доцільно зробити (з поміж названих технологій обираються такі, що доступні авторам проекту та є наявними на ринку).

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів [48].

Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 5.4).

Таблиця 5.4. — Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3

Таблиця 5.4.(продовження) — Попередня характеристика потенційного ринку стартап-проекту

2	Загальний обсяг продаж, грн/ум.од	300 грн
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	50 %

Середня норма рентабельності в галузі (або по ринку) порівнюється із банківським відсотком на вкладення. За умови, що останній є вищим, можливо, має сенс вкласти кошти в інший проект.

За результатами аналізу таблиці робиться висновок щодо того, чи є ринок привабливим для входження за попереднім оцінюванням [49].

Надалі визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (таблиця 5.5).

Таблиця 5.5. — Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Надання знання випускникам вузам, які потребує ринок інформаційних технологій сучасності	Вузи, компанії – роботодавці, студенти	Компанії заключають довготривалі договори, а стартапери віддають перевагу пробному терміну	стабільність роботи; невисока ціна; наявність випробувального періоду; наявність документації; підтримка необхідних платформ оптимізований час;

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (таблиці 5.6-5.7).

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку. Аналіз пропозиції необхідно виконати аналізуючи існуючі види конкуренцій.

Необхідною умовою ефективного функціонування механізму саморегулювання ринкової економіки є конкуренція. Вона є важливою рушійною силою розвитку ринкової економічної системи. Конкуренцію породжують об'єктивні умови ринкового господарювання: різні форми власності на засоби виробництва; повна економічна відокремленість і свобода вибору господарської діяльності товаровиробників, їх повна економічна залежність від кон'юнктури ринку; боротьба за джерела сировини, ринки збуту виробленої продукції, сфери використання капіталу з метою отримання найбільшого прибутку.

Конкуренція - це суперництво (змагальність) між різними учасниками ринкової економіки за найбільш вигідні умови виробництва та реалізації товарів і послуг, за привласнення найбільшого прибутку. Вона виступає силою, яка мобілізує особистий економічний інтерес і підприємницький потенціал та спрямована на їх максимальну реалізацію.

У Законі України "Про обмеження монополізму та недопущення недобросовісної конкуренції у підприємницькій діяльності" зазначається: конкуренція - це "змагальність підприємців, коли їх самостійні дії обмежують можливості кожного з них впливати на загальні умови реалізації товарів на ринку і стимулюють виробництво тих товарів, яких потребує споживач". Боротьба на ринку великої кількості різних товаровиробників і постачальників ресурсів за споживача (покупця) і економічний успіх - об'єктивний економічний закон.

Захист конкуренції, суб'єктів господарювання і споживачів від недобросовісної конкуренції передбачає демонополізацію вітчизняної економіки і створення ринкового конкурентного середовища.

Таблиця 5.6. — Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Підходить для нових проектів	Потребує визначеної структури бази даних	Імпорт схеми бази даних
2	Власний формат зберігання	При необхідності потрібна розробка сервісу преведення до визначеного формату	Додавання можливості автоматизованого експорту в різні типи сховищ, розробка додаткового ПЗ
3	Обмеженість функцій	Інструмент обмежений наявними функціями і не має деяких функцій, які мають конкуренти	Додавання нових функцій за потреби

Таблиця 5.7. — Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Незалежність від платформи	Можна використовувати як web інтерфейс, так і мобільний	Вихід на мобільний ринок вихід на рівень web додатків
2	Недоліки в існуючих альтернативах	Існуючі альтернативи або працюють повільно, або не є орієнтованими на конкретну предметну область	Модифікація існуючих платформ

Аналіз пропозицій зображено на таблиці.

Таблиця 5.8. — Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - монополія/олігополія/ монополістична/чиста	чиста	Прямі договори з стартапами, презентація продукту на виставках
2. За рівнем конкурентної боротьби - локальний/національний...	національний	Публікація статей на міжнародних сайтах
3. За галузевою ознакою - міжгалузева/ внутрішньогалузева	внутрішньогалузева	Розвивати напрямки систем формування сценаріїв
4. Конкуренція за видами товарів: - товарно-родова - товарно-видова - між бажаннями	товарно-видова	Розповідати про свої переваги перед конкурентом у цій галузі
5. За характером конкурентних переваг - цінова / нецінова	нецінова	Надання функцій, які не надають конкуренти, оптимізація функцій, що мають конкуренти
6. За інтенсивністю - марочна/не марочна	марочна	Надання функцій, які не надають конкуренти, оптимізація функцій, що мають конкуренти

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (Таблиця 5.9).

На основі аналізу конкуренції, проведеного в п. 3.5 (таблиця 5.9) [50], а також із урахуванням характеристик ідеї проекту (таблиця 5.2), вимог споживачів до товару (таблиця 6.5) та факторів маркетингового середовища (таблиця 5.6-5.7) визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлюється за таблицею 10

Таблиця 5.9. — Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
		Системе формування сценаріїв	Мінімізація витрат часу постачальників	Контроль якості	Лояльність споживачів
Висновки:	Визначити інтенсивність конкурентної боротьби з боку прямих конкурентів	Є можливості виходу на ринок, оскільки існуючі рішення не надають потрібних переваг	Постачальники підлаштовуються під ринок	Клієнти диктують вимоги згідно з умовами експлуатації	Обмеження для роботи на ринку через товари-замінники

Таблиця 5.10. — Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Орієнтація на предметну область формування сценаріїв розвитку електроенергетики	Існуючі конкуренти або не враховують особливості формування сценаріїв, або виконують процес побудови не оптимально

За визначеними факторами конкурентоспроможності (таблиця 5.10) проводиться аналіз сильних та слабких сторін стартап-проекту (таблиця 5.11)

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу [51] (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (таблиця 6.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (Таблиця 5.11).

Таблиця 5.11. — Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з Database Generator (даним продуктом)						
			-3	-2	-1	0	1	2	
1	Орієнтація на предметну область формування сценаріїв з розвитку електроенергетики	20	+						

Таблиця 5.12. — SWOT-аналіз стартап-проекту

<p>Сильні сторони:</p> <p>Актуальність користування системою, яка викликана бажанням розвитку електроенергетики</p> <p>Оцінка проходить відразу для великої кількості людей, а також у будь-який період часу.</p> <p>Актуальність користування системою, яка викликана постійним пошуком співробітників</p> <p>Невелика ціна користування за місяць</p>	<p>Слабкі сторони:</p> <p>Потребує масштабної рекламної компанії</p> <p>Орієнтація на інтернет, яка може відсіяти «не розвинутих» в технічному плані клієнтів</p> <p>Дороге зберігання великої к-сті даних</p> <p>Обробка даних</p>
<p>Можливості:</p> <p>Можливе продовження розробки проекту за кордоном, тому що проблема розвитку електроенергетики актуальна не лише в Україні</p> <p>Систему можна використати на ринку фрілансу, для відсіювання некомпетентних виконавців</p> <p>Можливість перегляду імовірних рішень</p> <p>Збереження результатів у різних форматах</p> <p>Зручність у використанні</p>	<p>Загрози:</p> <p>Відсутність користувачів через погану рекламну компанію</p> <p>Неможливість достукатися до необхідних API</p> <p>Втрата конфіденційних даних.</p>

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами)

впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Таблиця 5.13. — Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Орієнтація поточної моделі на ринок стартаперів	25 %	8 год
2	Орієнтація поточної моделі на ринок державних установ	20 %	72 год
3	Орієнтація поточної моделі на ринок ентерпрайз	35 %	168 год
4	Переорієнтація на розробку серверної частини	75 %	120 год
5	Переорієнтація на веб-розробку	45 %	96 год

Альтернатива, де отримання ресурсів є більш простим та ймовірним – №4 " Переорієнтація на розробку серверної частини ", що становить 75 відсотків. Це значення перевищує інші альтернативи.

Альтернатива, де строки реалізації є більш стислими – №2 " Орієнтація поточної моделі на ринок державних установ ". Терміни реалізації в цьому разі становлять 72 годин.

5.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (таблиця 4.14).

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку.

Таблиця 5.14. — Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Стартапери	Готові	Високий	Висока	Просто
2	Державні установи	Потребують недовгих переговорів	Середній	Середня	Складно

Таблиця 5.14 (продовження) — Вибір цільових груп потенційних споживачів

3	Ентерпрайз	Потребують довгих переговорів	Низький	Низька	Дуже складно
Які цільові групи обрано: стартапери					

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (таблиця 5.15).

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Таблиця 5.15. — Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
Орієнтація поточної моделі на ринок стартаперів	Стратегія концентрованого маркетингу	Стартапери потребують швидкості розробки, яку надає підтримка декількох платформ даним продуктом	Стратегія спеціалізації (спирається на диференціацію)

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту.

Наступним кроком є вибір стратегії конкурентної поведінки (таблиця 5.16).

Таблиця 5.16. — Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів	Чи буде компанія копіювати основні характеристик конкурента	Стратегія конкурентної поведінки
Ні	Шукати нових споживачів, забирати існуючих конкурентів		Стратегія заняття конкурентної ніші

5.5 Аналіз ринкових можливостей запуску стартап-проекту

Для цього у таблиці 5.17 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 5.17. — Визначення ключових переваг концепції потенційного товару

Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
Пришвидшення оптимальності роботи алгоритму	Побудова оптимального формування сценарію за оптимальний час	Конкуренти або не мають орієнтованос на електроенергетику, або формують сценарії не оптимальним шляхом

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його

надання (таблиця 5.18).

М/Нм – монотонні або немонотонні;

Вр/Тх/Тл/Е/Ор – вартісні, технічні, технологічні, ергономічні або органолептичні (останній – для продуктів харчування)

Після формування маркетингової моделі товару слід особливо відмітити – чим саме проект буде захищено від копіювання.

Захист може бути організовано за рахунок захисту ідеї товару (захист інтелектуальної власності), або ноу-хау, чи комплексне поєднання властивостей і характеристик, закладене на другому та третьому рівнях товару.

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (таблиця 5.29).

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (таблиця 5.20):

1. Проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту).
2. Вибір та обґрунтування оптимальної глибини каналу збуту.
3. Вибір та обґрунтування виду посередників.

Таблиця 5.19. — Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	1500...3000 грн	1800...4000 грн	25000...50000 грн	200...500 грн

Таблиця 5.20. — Формування системи збуту

№ п/п	Специфіка поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Клієнт повинен надаватися в режимах “тріал” та “повний” сплатити після закінчення випробувального строку	Легкість в встановленні, легкість в сплаті послуг	Веб-сайт	Проводити збут силами посередника формування сценаріїв

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 5.22).

Таблиця 5.21. — Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікації, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Купляють програми через авторизовану мережу	Веб-сайти	Формування сценарію розвитку	Довести, що програмний продукт оптимально формує сценарій	Формування сценарію розвитку

5.6 Висновки до розділу 5

Розроблений програмний продукт має переваги над існуючими конкурентами та є конкурентоздатним на ринку. Програма має шляхи подальшого розвитку, визначені маркетингові стратегії та шляхи збуту. Основна цільова аудиторія – працівники вузу та роботодавці в сфері інформаційних технологій.

ВИСНОВКИ

Отже, після виконання магістерської дисертації можна зробити наступні висновки:

В процесі аналізу ринку інформаційних технологій було виявлено, що існує відчутний дисбаланс між наявними знаннями випускників вищих навчальних закладів та потреб ринку інформаційних технологій. Немає сформованих навиків для спеціалізацій. Виявлено, що якість навчання в вузах оцінюється кількістю працевлаштованості студентів по спеціальності. Нажаль, такі результати не втішні. Самі випускники повідомляють, що вважають знання отримані в вузі є недоцільними.

Враховуючи вищезазначені проблеми було вирішено, що необхідно розробити ситсему корегування навчального плану в залежності від потреб ринку інформаційних технологій.

Реалізовані наступні завдання:

- створення веб-додатку з інтуїтивно зрозумілим інтерфейсом
- об'єднані існуючий функціонал та розроблений новий
- створено алгоритм порівняння навиків випускника та потреб вакансії.
- проведений порівняльний аналіз існуючих ресурсів та систем
- розроблений алгоритм оцінювання важливості навиків
- реалізований алгоритм порівняння існуючих знань та потреб ринку інформаційних технологій
- зберігання великої кількості даних
- організована статистика по потребам ринку інформаційних технологій
- організований пошук кандидатів та вакансій по заданим критеріям
- перегляд профілю випускника вузу.

Використання системи корегування навчального плану в залежності від потреб ринку інформаційних технологій дозволить створити профіль спеціаліста,

який буде здатний швидко адаптуватися та самостійно набирати необхідні знання. За допомогою системи можна буде слідкувати за тенденціями ринку інформаційних технологій і вчасно проводити корегування навчального плану у відповідності до потреб.

Такі дії допоможуть вирішити проблему з працевлаштування випускників вузів. Допоможе залучити більше інвестицій в вузи.

Впровадження системи корегування навчального плану в залежності від потреб ринку інформаційних технологій забезпечить випускників вузів підтримку в працевлаштуванні, вузу – дасть можливість залучення нових інвестицій для підвищення якості навчання, роботодавцям – отримувати працівників з релевантними знаннями.

Потенційними користувачами програмного продукту є рекрутери, працівники вузів і студенти.

При розробці системи корегування навчального плану в залежності від потреб ринку інформаційних технологій я використала інтегровану систему розробки Visual Studio, .NET фреймворк, RAZOR, CSS, Javascript, AJAX.

Розроблений програмний продукт має переваги над схожими застосунками, які представлені на ринку. Він забезпечений більш зрозумілим інтерфейсом, завдяки мінімальній кількості функцій, забезпечена швидкодія, дешевший, в порівнянні з аналогами.

Функціонування запропонованого програмного продукту та його роль в системі освіти слід розглядати як початковий процес, здатний допомогти змінити ситуацію в навчанні спеціалістів в сфері інформаційних технологій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Черняк Л. Большие Данные — новая теория и практика (рус.) //Открытые системы. СУБД. — М.: Открытые системы, 2011. — № 10
2. Шполянская И. Ю., Воробьева А. М. Модели и методы оптимизации структуры образовательных порталов вузов в системе интернет-маркетинга //Вестник Ростовского государственного экономического университета (РИНХ). 2012. № 37. С. 311.
3. Національний класифікатор України. Класифікатор професій ДК 003:2010 [Електронний ресурс]. – Режим доступу: http://hrliga.com/index.php?module=norm_base&op=view&id=433.
4. Научук О. Визначення професійних якостей майбутніх інженерів-програмістів у галузі комп'ютерних мереж / О. Научук // Наук. вісн. Мелітопольського держ. пед. ун-ту. Сер. Педагогіка. – 2015. – № 1 (14). – С. 358.
5. Русанова Е. И. О востребованности молодого специалиста в сфере ИТ на современном рынке труда / Е. И. Русанова // ПНИО. – 2013. – № 5. – С.224
6. Шевчук С. В. Українська мова за професійним спрямуванням : підручник / С. В. Шевчук, І. В. Клименко. – Київ : Алерта, 2011 – 696 с.
7. Шполянская И. Ю. Анализ требований работодателей в системе поддержке трудоустройства выпускников вуза / И. Ю. Шполянская, Н. Ю. Мисиченко // Вестник Ростовского гос. эконом.ун-та «РИНХ». – 2009. – № 2. – С. 550.
8. Ильчук С. Б. Востребованность выпускников на рынке труда как индикатор эффективной деятельности вузов //Журнал социологии и социальной антропологии 2008. Т XI #1
9. Хубаев Г.Н. Методика экономической оценки потребительского качества программных средств //Программные продукты и системы (SOFTWARE and SYSTEMS). – 1995. – N 1. – С. 250
10. Качалов в.а. Проблемы управления качеством в вузах // Стандарты и качество. 2000. № 10, с. 96- 100.

11. Пучков Н.П., Дворецкий С.И., Таров В.П. Научно-методические аспекты обеспечения качества и инновационной деятельности технического вуза машиностроительного профиля. - М.: "Изд-во машиностроение-1". 2004. - 184 с.
12. Шполянская И. Ю., Мисиченко Н. Ю. Анализ требований работодателей в системе поддержки трудоустройства выпускников вуза // Вестник Ростовского государственного экономического университета «РИНХ». 2009. No 28. С.337.
13. Пашін В.П. Методичні вказівки до виконання економіко-організаційного розділу дипломних проектів (робіт) бакалаврів і спеціалістів для студентів Інституту прикладного системного аналізу з дисципліни “Економіка та організація виробництва” для студентів спеціальностей 7.080204 - “Соціальна інформатика”, 7.080203 - “Системний аналіз і управління”, 7.080404 – “Інтелектуальні системи прийняття рішень”, 7.080402 - “Інформаційні 75 технології проектування” / Пашін В.П., Романов В.В., Єгорова Н.В. – К. : НТУУ “КПІ”, 2011. – 118 с.
14. Гаврилова Т.А., Хорошевский В.Ф., 2000. Базы знаний интеллектуальных систем / Учебник для вузов. – СПб, Изд-во “Питер”, 2000
15. Аляев Ю.А., Беляков А.Ю., Гейхман Л.К. Формирование управленческих компетенций с использованием экспертных систем обучения // Открытое образование. 2006. - №3. - С.195.
16. Болотов, В.А. Компетентностная модель: от идеи к образовательной программе// Перемены : Пед. Журн. 2004. - N2. - С. 180.
17. Бороздина Л.И. Рыночные ориентиры студенчества: (По данным социологического мониторинга) // Социально-экономические условия и методы эффективной коммерческой деятельности. Иркутск, 2000. - С. 156
18. Вендров А.М. Case-технологии. Современные методы и средства проектирования информационных систем. М.: Финансы и статистика, 1998. -176 с
19. Дорофеев А. Профессиональная компетентность как показатель качества образования // Высш. Образование в России. 2005. - №4. - С. 125.
20. Джонс Дж. К. Методы проектирования. М.: Мир, 1986. - 326 с.
21. Качалов В.А. Стандарты ИСО 9000 и проблемы управления качеством в вузах (записки менеджера качества). М.: ИздАТ, 2001. -128 с.

22. Кирсанов А.А. Методологические проблемы создания прогностической модели специалиста. Казань: КГТУ, 2000. - 228 с.
23. Кравец А.Г., Обухов А.С. «Автоматизированная система проверки профессиональных знаний специалистов на предприятии». Вестник компьютерных и информационных технологий. 2007 - № 3.
24. Роберт Э. Уолтерс, Майкл Коулс. SQL Server 2008: ускоренный курс для профессионалов = Accelerated SQL Server 2008. — М.: «Вильямс», 2008. — С. 768.
25. Гради Буч, Роберт А. Максимчук, Майкл У. Энгл, Бобби Дж. Янг, Джим Коналлен, Келли А. Хьюстон. Объектно-ориентированный анализ и проектирование с примерами приложений. — 3-е издание. — «Вильямс», 2010.
26. Лука Карделли, Peter Wegner On Understanding Types, Data Abstraction, and Polymorphism // ACM Computing Surveys. — New York, USA: ACM, 1985. — Т. 17, вып. 4. — С. 471–523.
27. Антони Синтес. Освой самостоятельно объектно-ориентированное программирование за 21 день = Sams Teach Yourself Object-Oriented Programming in 21 Days. — М.: «Вильямс», 2002. — С. 672.
28. Rodgers, Peter. "Service-Oriented Development on NetKernel- Patterns, Processes & Products to Reduce System Complexity". CloudComputingExpo. SYS-CON Media. Retrieved 19 August 2015.
29. Роб Камерон, Дэйл Михалк ASP.NET 3.5, компоненты AJAX и серверные элементы управления для профессионалов = Pro ASP.NET 3.5 Server Controls with AJAX Components. — М.: «Вильямс», 2009. — С. 608.
30. Стивен Сандерсон ASP.NET MVC Framework с примерами на C# для профессионалов = Pro ASP.NET MVC Framework. — М.: «Вильямс», 2010. — С. 560
31. Даниэль Арсеновски Рефакторинг в C# и ASP.NET для профессионалов = Professional Refactoring in C# & ASP.NET. — М.: «Диалектика», 2009. — С. 528
32. Дэниел Вулстон. Ajax и платформа .NET 2.0 для профессионалов = Pro Ajax and the .NET 2.0 Platform. — М.: Вильямс, 2007. — С. 464.

33. Стив Резник, Ричард Крейн, Крис Боуэн. Вступление // Основы Windows Communication Foundation для .NET Framework 3.5 = Windows Communication Foundation for .NET Framework 3.5 / Пер. с англ. Слинкина А. А. — ДМК Пресс. — М., 2008. — С. 16.
34. Адам Фримен. ASP.NET MVC 4 с примерами на C# 5.0 для профессионалов, 4-е издание = Pro ASP.NET MVC 4, 4th edition. — М.: «Вильямс», 2013. — 688 с.
35. Стивен Хольцнер. Ajax Библия программиста = Ajax Bible. — М.: Диалектика, 2009. — С. 553.
36. Джесс Чедвик и др. ASP.NET MVC 4: разработка реальных веб-приложений с помощью ASP.NET MVC = Programming ASP.NET MVC 4: Developing Real-World Web Applications with ASP.NET MVC. — М.: «Вильямс», 2013. — 432 с.
37. JavaScript. Библия пользователя = JavaScript. Bible / Денни Гудман (Danny Goodman), Майкл Моррисон (Michael Morrison); пер. с англ. И. В. Василенко. — 5-е
38. Alexei White. Major JavaScript Engines // JavaScript Programmer's Reference. — Indianapolis, IN 46256: Wiley Publishing, Inc., 2009. — P. 12 — 13
39. Дейв Крейн, Эрик Паскарелло, Даррен Джеймс. AJAX в действии: технология — Asynchronous JavaScript and XML = Ajax in Action. — М.: Вильямс, 2006. — С. 640. — ISBN 1-932394-61-3.
40. JavaScript. Карманный справочник. Необходимый код и команды = JavaScript. Phrasebook. Essential code and commands / Кристиан Уэнц (Cristian Wenz); пер. с англ. И. В. Берштейн. — Москва, Санкт-Петербург, Киев: ООО "И. Д. Вильямс", 2008. — С. 18. — 272 с.
41. Дейв Крейн, Бер Бибо, Джордон Сонневельд. Ajax на практике = Ajax in Practice. — М.: Вильямс, 2007
42. Б. Маклафлин. Изучаем Ajax = Head Rush Ajax. — СПб.: Питер, 2007

43. Роберт Виейра. Программирование баз данных Microsoft SQL Server 2005. Базовый курс = Beginning Microsoft SQL Server 2005 Programming. — М.: «Диалектика», 2007. — С. 832
44. Майк Гандерлой, Джозеф Джорден, Дейвид Чанц. Освоение Microsoft SQL Server 2005 = Mastering Microsoft SQL Server 2005. — М.: «Диалектика», 2007. — С. 1104.
45. Microsoft® SQL Server™ 2005. Реализация и обслуживание. Учебный курс Microsoft (Экзамен 70-431). — М.: «Питер», 2007. — С. 767.
46. Петкович Д. Microsoft SQL Server 2008. Руководство для начинающих. — С.: БХВ-Петербург, 2009. — С. 752
47. Бариленко В. И. Бизнес-анализ как важный вид консалтинговых услуг // РИСК: Ресурсы, Информация, Снабжение, Конкуренция. — № 4. — 2012. — С.202-207.
48. Калянов Г.Н. Теория и практика реорганизации бизнес-процессов. — М.: СИНТЕГ, 2000. 212 с.
49. Кватрани Т. Rational Rose 2000 и UML. Визуальное моделирование: Пер. с англ. М.: ДМК Пресс, 2001.-176 с.
50. Квашнин А. Как продвигать проекты коммерциализации технологий: серия методических материалов «Практические руководства для центров коммерциализации технологий» / М. Катешова, А. Квашнин, под рук. П. Линдхольма, проект EuropeAid «Наука и коммерциализация технологий», 2006. – 52 с.
51. Харниш, В. Правила прибыльных стартапов : как расти и зарабатывать деньги / В. Харниш ; пер. с англ. В. Хозинского. – Москва : Манн, Иванов и Фербер, 2012. – 279 с.