

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

**Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій**

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 20\_\_ р.

**Дипломний проєкт**  
**на здобуття ступеня бакалавра**  
**за освітньо-професійною програмою «Інтегровані інформаційні системи»**  
**спеціальності 126 «Інформаційні системи та технології»**  
**на тему: «Адаптивна платформа для персоналізованого онлайн-навчання»**

Виконав (-ла):

студент (-ка) IV курсу, групи ІА-94

Сук Денис Олегович \_\_\_\_\_

Керівник:

Асистент каф. ІСТ,

Шинкевич Микола Костянтинович \_\_\_\_\_

Рецензент:

Доцент кафедри ОТ, кандидат технічних наук

Верба Олександр Андрійович \_\_\_\_\_

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студент (-ка) \_\_\_\_\_

Київ – 2023 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра інформаційних систем та технологій**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

« \_\_\_\_ » \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

**Суку Денису Олеговичу**

1. Тема проєкту «Адаптивна платформа для персоналізованого онлайн-навчання», керівник проєкту Шинкевич Микола Костянтинович, Асистент кафедри ІСТ, затверджені наказом по університету від «31» травня 2023 р. № 2101-с.
2. Термін подання студентом проєкту: «12» червня 2023 р.
3. Вихідні дані до проєкту: документація, діаграми, опис роботи програмного забезпечення, опис розгортання системи.
4. Зміст пояснювальної записки: Опис предметної області, аналіз сучасних існуючих рішень, аналіз використаних технологій, розробка власної моделі для представлення архітектури програмного забезпечення, опис роботи програми.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): функціональна схема (діаграма класів), принципова схема (діаграма процесів), структурна схема системи, схема варіантів використання.

6. Дата видачі завдання 1 лютого 2023 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Затвердження теми роботи	22.02.2023	
2	Вивчення та аналіз завдання	28.02.2023	
3	Розробка архітектури та загальної структури системи	15.03.2023	
4	Розробка структур окремих частин системи	31.03.2023	
5	Програмна реалізація системи	20.04.2023	
6	Оформлення пояснювальної записки	01.06.2023	

Студент

Денис СУК

Керівник

Микола ШИНКЕВИЧ

## АНОТАЦІЯ

Сук Д.О. Адаптивна платформа для персоналізованого онлайн-навчання. КПІ ім. Ігоря Сікорського, Київ, 2023.

Проект містить 65 с. тексту, 32 рисунки, посилання на 15 літературних джерел, додатки та 4 конструкторські документи.

Ключові слова: управління навчанням, система керування, Java MVC, адаптивна платформа, онлайн-навчання.

Дипломна робота присвячена розробці адаптивної платформи для персоналізованого онлайн-навчання. В сучасному світі освіта стає все більше цифровою, а онлайн-навчання є ефективним та доступним засобом отримання знань. Однак, багато існуючих платформ не забезпечують достатньої персоналізації, що може призводити до недосягнення навчальних цілей та зниження мотивації учнів. Метою цього проекту є створення платформи, яка забезпечує індивідуальну підготовку та персоналізоване навчання для кожного учня. Для реалізації проекту використовується сучасні технології веб-розробки, бази даних та аналізу даних. Платформа розроблена з урахуванням зручного та інтуїтивно зрозумілого інтерфейсу для учнів та викладачів.

## SUMMARY

Suk D.O. Adaptive Platform for Personalized Online Learning. Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2023.

The project consists of 65 pages of text, 32 pictures, references to 15 literary sources, appendices, and 4 design documents.

Keywords: learning management, control system, Java MVC, adaptive platform, online learning.

The diploma thesis is dedicated to the development of an adaptive platform for personalized online learning. In the modern world, education is increasingly becoming digital, and online learning is an effective and accessible means of acquiring knowledge. However, many existing platforms do not provide sufficient personalization, which can lead to the failure to achieve learning goals and a decrease in student motivation. To implement the project, modern web development technologies, databases, and data analysis are utilized. The platform is designed with a convenient and intuitive interface for both students and instructors.

Номер рядка	Формат		Найменування	Кільк. аркушів	Номер екзем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5						
6	A4	IA94.190BAK.003 ПЗ	Пояснювальна записка	65		
7						
8	A3	IA94.190BAK.003 Д1	Адаптивна платформа для	1		
9			персоналізованого			
10			онлайн-навчання.			
11			Діаграма класів			
12	A3	IA94.190BAK.003 Д2	Адаптивна платформа для	1		
13			персоналізованого			
14			онлайн-навчання.			
15			Діаграма діяльності			
16	A3	IA94.190BAK.003 Д3	Адаптивна платформа для	1		
17			персоналізованого			
18			онлайн-навчання.			
19			Структурна схема системи			
20	A3	IA94.190BAK.003 Д4	Адаптивна платформа для	1		
21			персоналізованого			
22			онлайн-навчання.			
23			Діаграма прецедентів			
24						
25						
26						
27						
28						

					<b>IA94.190BAK.003 ВП</b>		
Зм.	Аркуш	№ докум.	Підпис	Дата			
Розроб.		Сук Д.О.			Літ.	Аркуш	Аркушів
Керівн.		Шинкевич Н.К.			Т	1	1
					КПІ ім. Ігоря Сікорського Група ІА-94		
Затв.							

Адаптивна платформа для  
персоналізованого онлайн-  
навчання.  
Відомість проєкту

**Пояснювальна записка  
до дипломного проєкту  
на тему: «Адаптивна платформа для персоналізованого  
онлайн-навчання»**

Київ – 2023 року

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ .....	4
ВСТУП .....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	6
1.1 Історія перших адаптивних платформ для онлайн-навчання .....	6
1.2 Платформа Moodle .....	9
1.3 Платформа Classroom .....	11
1.4 Платформа Teachmint .....	14
Висновок до розділу 1 .....	17
2 АНАЛІЗ ВИКОРИСТАНИХ ТЕХНОЛОГІЙ.....	18
2.1 Мова програмування Java .....	18
2.2 Використані фреймворки .....	20
2.2.1 Spring Boot.....	21
2.2.2 Spring Web MVC .....	23
2.2.3 Spring Security .....	27
2.2.4 Hibernate .....	30
2.3 MySQL.....	32
2.4 Swagger .....	36
2.5 JavaScript .....	38
Висновок до розділу 2 .....	39
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	40
3.1 Структура проєкту.....	40
3.2 Автентифікація .....	43
3.3 Розробка структури бази даних .....	45
3.4 Серверна частина.....	47

					<b>IA94.190BAK.003 ПЗ</b>			
		№ докум.	Підпис	Дата	Адаптивна платформа для персоналізованого онлайн-навчання Пояснювальна записка	Літ.	Аркуш	Аркушів
Розробив	Сук Д.О.					2	65	
Перевірив	Шинкевич Н.К.							
Н. Контр.						КПІ ім. Ігоря Сікорського		
Затвердив					ФІОТ, IA-94			

3.5 Клієнтська частина.....	51
Висновок до розділу 3 .....	53
4 РОЗГОРТАННЯ СЕРВІСУ ТА ДЕМОНСТРАЦІЯ .....	54
4.1 Розгортання сервісу .....	54
4.2 Тестування роботи застосунку .....	56
Висновок до розділу 4 .....	62
ВИСНОВОКИ .....	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	64

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

## ПЕРЕЛІК СКОРОЧЕНЬ

ПК – комп'ютер

БД – база даних

JDK (Java Development Kit) – набір інструментів для розробників

JVM (Java Virtual Machine) – віртуальна машина Java

CMS (Content Management System) – система управління вмістом

CRM (Customer Relationship Management) – система управління взаємовідносинами з клієнтами

MVC ( Model-View-Controller) – архітектурний шаблон програмування програмного забезпечення

HTTP (Hypertext Transfer Protocol) – протокол передачі гіпертексту

API (Application Programming Interface) – програмний інтерфейс

ORM (Object-Relational Mapping) – зв'язок БД з мовою програмування

					ІА94.190БАК.003 ПЗ	Арк.
Зм	Арк.	№ докум.	Підпис	Дата		4

## ВСТУП

В наш час велика кількість навчальних закладів зіштовхнулись з проблемою проведення занять в аудиторіях університетів та шкільних класів. Починаючи з 2020 року в умовах пандемії школи та університети вийшли на дистанційне навчання і по цей час університети знаходяться на дистанційній формі навчання. На тлі цієї проблеми було запропоновано чимало програмних додатків для більш зручного викладення матеріалів навчання, лекційних матеріалів та різних лабораторних робіт, але як такого універсально і зручного додатку не було запропоновано.

Розробка даної системи допоможе вирішити деякі незручності, які наявні в інших подібних системах, в додатку будуть реалізовані функції які будуть надзвичайно корисні і після того, як навчання повернеться до звичного формату. Наприклад, вчителю не потрібно буде вести журнал відвідувань і носити його на кожне лекційне чи практичне завдання, достатньо буде мати телефон або ПК.

Однією з основних переваг системи буде можливість ведення журналу відвідуваності онлайн. Кожен учень буде мати власний обліковий запис, через який він зможе перевірити свої оцінки та подивитися розклад. Це знизить навантаження на вчителів та спростить процес обліку відвідування. Вчителі зможуть легко створювати тести та контрольні завдання, а студенти зможуть зручно їх виконувати та отримувати результати.

Окрім цього, система забезпечить можливість управління класами та періодами навчання. Вчитель зможе легко перенести дані учнів для наступного навчального року без додаткових змін у інших журналах. Також вчитель буде мати змогу у редагуванні поточних навчальних матеріалів для забезпечення постійної їх актуалізації.

В підсумку, можна сказати, що обрана тема є досить корисною для сучасного навчання, розробка універсальної системи для навчання та взаємодії між вчителями та студентами допоможе вирішити проблеми, з якими стикаються багато навчальних закладів під час проведення навчального процесу.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Історія перших адаптивних платформ для онлайн-навчання

Компанія Knewton була заснована у 2008 році Хосе Феррейра, колишнім керівником компанії Kaplan, Inc, щоб дати можливість школам, видавцям та розробникам забезпечити адаптивне навчання для кожного учня. У Knewton вважають, що немає двох однакових учнів, які б мали однаковий досвід або стилі навчання, і що освіта повинна бути адаптована до сильних і слабких сторін кожної дитини. Щоб порекомендувати наступний найкращий курс/заняття для учня, який максимально сприятиме його/її навчанню, Кноутон спирається на історію кожного учня, на інтереси учнів зі схожими стилями навчання та на десятиліття досліджень, присвячених покращенню навчального досвіду.

Таким чином, Knewton допоміг Університету штату Арізона (серед іншого) підвищити рівень успішності на 17%, зменшити кількість відмов від курсу на 56% та прискорити навчання, оскільки 45% студентів закінчили курс на 4 тижні раніше.

Knewton використовує технологію адаптивного навчання для створення платформи, яка дозволяє навчальним закладам і видавцям програмного забезпечення адаптувати навчальний контент для особистого використання. Розпочавшись як програма для підготовки до онлайн-тестів, Knewton тепер має на меті визначити наступний найкращий крок у навчальній подорожі користувача.

Завдяки партнерству з провідними університетами США та видавництвами, такими як Pearson, адаптивна навчальна платформа має на меті покласти край універсальним навчальним програмам, роблячи персоналізовані навчальні плани доступними для учнів К-12 та студентів коледжів.

Рішення Knewton пропонує двосторонній підхід до рекомендацій щодо навчальної програми, який орієнтує студентів на те, що найкраще вивчати далі і як вони повинні це робити.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

Рекомендації можуть бути використані для управління повним навчальним процесом або можуть слугувати індивідуальними корекціями у відповідь на результати тестів.

Це досягається за допомогою даних. Як тільки студент входить на платформу, кожне натискання клавіш і рух миші записується як частина потоку кліків, щоб зрозуміти його поведінку.

Потім алгоритм адаптивного навчання використовує ці дані, щоб зрозуміти різні аспекти навчального процесу, такі як зацікавленість, рівень знань, нудьга і розчарування, що вимірюються через час, витрачений на навчальні модулі, рівень помилок, отримані оцінки тощо. Наприклад, Knewton використовує теорію відповідей на завдання для оцінки та порівняння рівня знань на основі відповідей особи на тести порівняно із загальними демографічними показниками учасника тестування.

Knewton вирішив позиціонувати себе як адаптивну навчальну платформу, яка співпрацює з постачальниками освітнього контенту для створення персоналізованого навчального досвіду. Серед їхніх партнерів - Houghton-Mifflin, Pearson і Triumph Learning, що надало їм значної ваги на ринку США. Крім того, вони обслуговують 13 мільйонів студентів по всьому світу через свою платформу, оскільки вони також орієнтуються на ринки, що розвиваються, де існує менше структурних освітніх ініціатив, які необхідно подолати. Нарешті, Knewton також працює над створенням партнерств з МВОК та університетами. Результати партнерства Knewton з Університетом Арізони у розробці курсів з математики для розвитку показують, що відсоток успішності збільшився на 11 відсоткових пунктів, тоді як відсоток відмов від навчання зменшився на 50%.

Серед конкурентів Knewton у сфері адаптивного навчання - Kidaptive, McGraw-Hill Education, австралійська компанія Smart Sparrow, Dreambox Learning та Desire2Learn. Хоча кожен з конкурентів має свій власний набір результатів і перемог, варто зазначити, що Smart Sparrow повідомила про зниження рівня

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

неуспішності з 31% до 7% на курсі механіки, і вони також співпрацюють з Університетом штату Арізона. Отже, хоча Knewton бачить багатообіцяючі результати своєї платформи, і хоча вони мають велику популярність, конкуренти здатні отримати подібні, якщо не кращі результати. Одним із псевдоконкурентів, з яким Knewton міг би подумати про партнерство, могли б стати альтернативні школи, такі як AltSchool, оскільки чартерні школи та альтернативні методи навчання стають дедалі популярнішими. Це дало б їм ще один шлях для використання своєї платформи, а також дало б їм перевагу над нинішніми конкурентами.

#### Пропоновані зміни

##### Де студенти сидять у класі:

–Knewton використовує моделювання залученості, щоб визначити, наскільки віртуальні студенти зацікавлені в навчанні. Цю ж методологію можна було б застосувати і в класі;

–використовуючи фотосенсори, Knewton може включити розташування сидінь у класі в свою аналітику. Можливо, можна було б визначити, чи впливає на навчання студента місце, де він сидить у класі відносно викладача та інших студентів.

##### Інтеграція зі стандартизованим тестуванням:

–адаптивна онтологія Knewton може бути використана для кращого розуміння готовності студентів до стандартизованого тестування та ефективності стандартизованого тестування. Зокрема, взаємозв'язок між оцінкою та необхідністю, який дає уявлення про розуміння студентами понять, що вимагають розуміння попередніх понять;

–інструмент Knewton може допомогти розробникам стандартизованих тестів довести, що концепції, призначені для тестування, дійсно є тими, що тестуються. Він також може допомогти студентам підготуватися до тесту.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

Інтеграція з андеррайтерами студентських кредитів:

–результати Університету штату Арізона свідчать про значне покращення показників відмов від кредитів. Незавершення навчальних програм є основною причиною дефолтів за студентськими кредитами. Інсайти Knewton можуть бути використані як індикатор ризику неповернення студентських кредитів;

–конфіденційність може бути проблемою на індивідуальному рівні. [1]

## 1.2 Платформа Moodle

Moodle - це веб-платформа для управління навчанням, яка використовується в освітніх установах для організації навчальних процесів та електронного навчання. Основною метою використання Moodle є підтримка і полегшення навчального процесу, спільної роботи вчителів і студентів, забезпечення доступу до навчального матеріалу та інструментів для взаємодії та оцінювання.

Moodle надає можливість створювати та організовувати навчальні курси, завантажувати навчальний матеріал та вивчати його в онлайн-режимі. Він також забезпечує спілкування та співпрацю між вчителями та студентами, включаючи форуми обговорень, чати, блоги та спільну роботу над завданнями та проектами.

Одним з важливих аспектів Moodle є можливість оцінювання знань студентів. Вчителі можуть створювати завдання, тести та опитування для перевірки рівня засвоєння матеріалу. Також, Moodle дозволяє вчителям стежити за прогресом студентів, генерувати звіти та оцінки.

Користувач має можливість зручно переглянути всі етапи навчання на сайті або в додатку, відвідавши відповідну вкладку або розділ. Цей список етапів може бути представлений у зручній інтерактивній формі, що дозволяє користувачеві швидко орієнтуватися та знайти потрібний предмет або тему.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

У вкладці зі списком етапів навчання користувач може бачити загальну структуру курсу або програми, розбиту на конкретні теми, модулі або уроки. Кожен етап може мати назву або опис, який надає короткий огляд того, що буде вивчатися в цьому етапі.

Однією з переваг Moodle є його гнучкість та розширюваність. Він є відкритою платформою з активною спільнотою розробників, що дозволяє вносити зміни та доповнення до функціональності платформи відповідно до потреб освітньої установи.

Загалом, Moodle є гарним інструментом для організації ефективного навчання, сприяє взаємодії між учасниками навчального процесу та забезпечує доступ до необхідних ресурсів та інструментів для успішного навчання та оцінювання студентів.

Звіт по користувачу у курсі - Сук Денис Олегович

[Переглянути звіт](#) Звіт по користувачу у курсі

Елемент оцінювання	Обрахована значимість	Оцінка	Інтервал	Відсоток	Відгук	Внесок у підсумок курсу
<b>Екологічна та природно-техногенна безпека</b>						
Тези доповідей на наукові конференції та наукові статті	0,00 % (Порожньо)	-	0-10	-		0,00 %
тест за курс	100,00 %	9,00	0-10	90,00 %		90,00 %
Загальне за курс	-	9,00	0-10	90,00 %		-

Рисунок 1.1 — Веб-сайт moodle.org, журнал оцінок

Кожен користувач має можливість отримати актуальні результати оцінювання для вибраного предмету календарного контролю та іншу відповідну інформацію, як показано на рисунку 1.1. Крім того, доступні лекції та інші матеріали для навчання.

Усі важливі дані виділяються спеціальними символами, щоб зробити їх зрозумілішими та зручнішими для користувача.

Але Moodle не надає змоги вносити оцінки за роботу на лекції або практичному завданні, оскільки він більш націлений на викладення матеріалів для лекцій чи практик, вчитель немає змоги вносити помітки чи був студент присутнім на парі, і за можливості поставити оцінку за продуктивну роботу. Вчитель повинен з декількох ресурсів виводити фінальну оцінку, що є не дуже зручним, опираючись на свій досвіді можу навести приклад використання excel та платформи Moodle, такий спосіб ведення контролю не є дуже зручним, оскільки, excel файл може загубитись або не зберегтись останні зміни, що призводить до неточності в оцінюванні. Тому було запропоновано саме цей застосунок.

### 1.3 Платформа Classroom

Google Classroom використовується як платформа для віддаленого навчання та організації класу. Його основна мета полягає у спрощенні процесу комунікації та обміну інформацією між вчителями і учнями.

Google Classroom був анонсований у травні 2014 року, попередня версія була доступна для певних передплатників G Suite for Education. Потім, 12 серпня 2014 року, він став загальнодоступним, а 14 січня 2015 року Google представив API Classroom і кнопку "Поділитися" для застосунків, що дозволило вчителям, розробникам та іншим шкільним адміністраторам взаємодіяти з ним ще глибше.

Крім того, Google інтегрував у Classroom Календар Google, який слугує нагадуванням про терміни виконання завдань та інші заходи на 2015 рік. Компанія офіційно представила продукт у травні того ж року.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

API Classroom і кнопка "Поділитися" для веб-сайтів були представлені Google 14 січня 2021 року, що дозволило вчителям, розробникам та іншим шкільним адміністраторам підключитися до продукту після того, як він був вперше оприлюднений 12 серпня 2020 року. Щоб допомогти учням відстежувати терміни виконання завдань та інші майбутні події, Google інтегрував у платформу Classroom календар Google Calendar.

Про успішність даної платформи можуть свідчити показники за 2019-2020 роки, коли через пандемію люди були вимушені перейти на дистанційний режим роботи та навчання, це зображено на рисунку 1.2. [2]

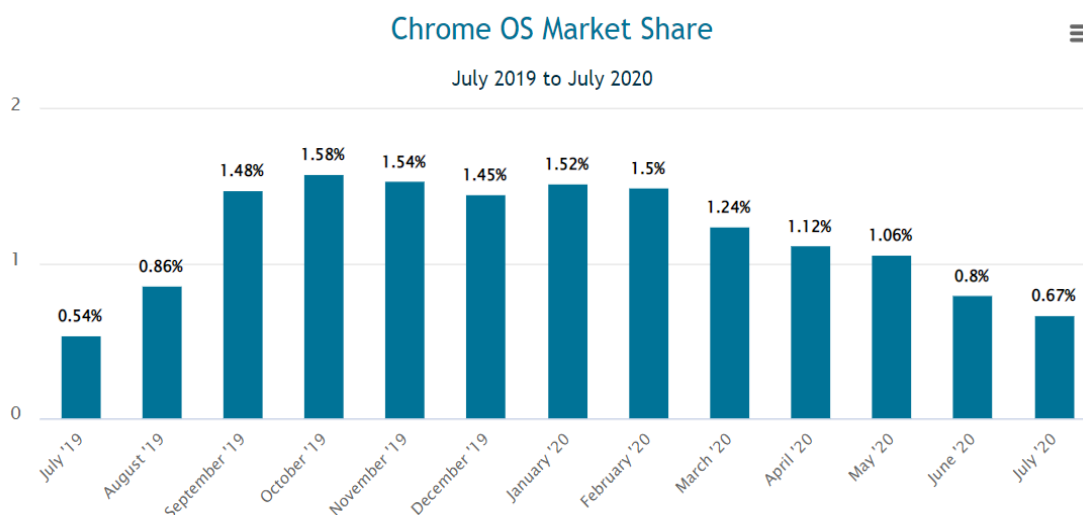


Рисунок 1.2 – Діаграма частини ринку Chrome

Основні функції та застосування Google Classroom включають передачу матеріалів та завдань. Вчителі можуть створювати віртуальні класи та завантажувати матеріали, такі як лекції, презентації, документи, відео чи посилання.

Також, використання Google Classroom дозволяє вчителям призначати завдання учням та встановлювати крайні терміни для їх виконання. Після виконання завдання, учні можуть надсилати їх назад для перевірки та оцінювання. Вчителі можуть надавати зворотний зв'язок та виставляти оцінки.

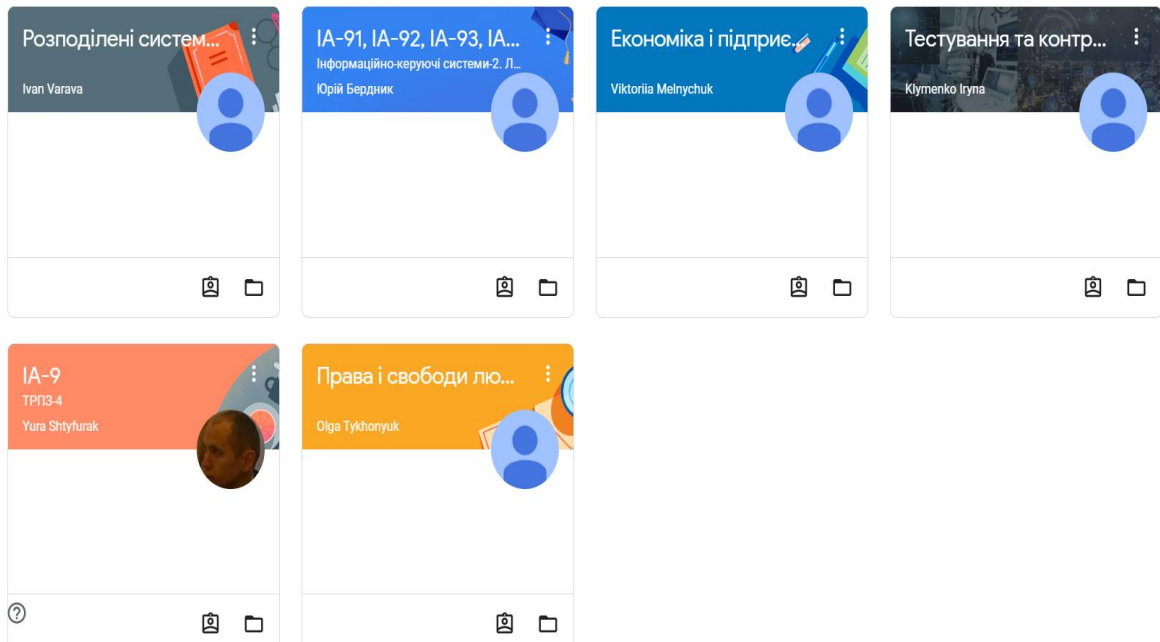


Рисунок 1.3 — Сервіс classroom.google.com. Головна сторінка з класами

Google Classroom також надає зручні засоби обміну повідомленнями та спілкування між вчителями та учнями. Через вбудований чат можна вирішувати питання, обговорювати завдання та надавати додаткову підтримку. Також вбудований чат підтримує обмін різними типами файлів, допомагає швидше узгоджувати зміни у завданні з вчителем. Також є можливість залишити відгук для учня.

На рисунку 1.3 можна побачити, що платформа дозволяє вчителям створювати розклад занять та надсилати нагадування учням про наближення термінів завдань або інших подій в класі. Це допомагає учням бути організованими та не пропускати важливі дедлайни. Крім того, Google Classroom інтегрована з іншими інструментами Google, такими як Google Документи, Google Презентації та Google Календар.

Підсумовуючи, можна виділити наступні можливості:

- студенти можуть відстежувати та надсилати класні роботи та завдання;
- учні можуть спілкуватися електронною поштою або через потоки класу;

- викладачі можуть ініціювати відеозустрічі, організовувати та адмініструвати заняття онлайн, усуваючи потребу в паперових позначках;
- вчителі можуть надавати негайний зворотній зв'язок і залучати учнів до розмов через навчальний потік;
- батьки можуть переглядати діяльність своїх дітей;
- батьки можуть отримувати електронною поштою підсумок роботи учня.

Зважаючи на всі ці переваги, африканським школам наполегливо рекомендується впроваджувати цю технологію у свій навчальний процес. [3]

Це дуже зручна платформа для опублікування матеріалів, лабораторних робіт та іншої інформація, яка стосується предмету, але один із недоліків, це відсутня можливість автоматизовано виводити оцінки, оскільки необхідно переносити всі оцінки в окремий журнал, який неаявний на цій платформі, а також учням, може буде незручно переглядати оцінки, оскільки потрібно заходити в різні завдання та дивитись оцінку за конкретну роботу, а не за всі одразу. Цю незручність буде вирішено у розробленому застосунку. Буде створенно онлайн журнал, де можна будет переглядати оцінки по обраному предмету та загальні оцінки учня. Також буде створенно зручне графічне зображення для покращення розуміння статистики кожного учня. За допомогою статистики у графічному відображені, можна буде швидко зрозуміти коли учень почав відставати від програми навчання.

#### 1.4 Платформа Teachmint

Застосунок використовується як платформа для віддаленого навчання та управління класом. Teachmint надає засоби для проведення онлайн-уроків, взаємодії вчителів і учнів, надання освітніх матеріалів та управління класом. Вчитель може проводити відеоуроки, ділитися презентаціями, призначати домашні завдання, вести журнал присутності та оцінки учнів.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

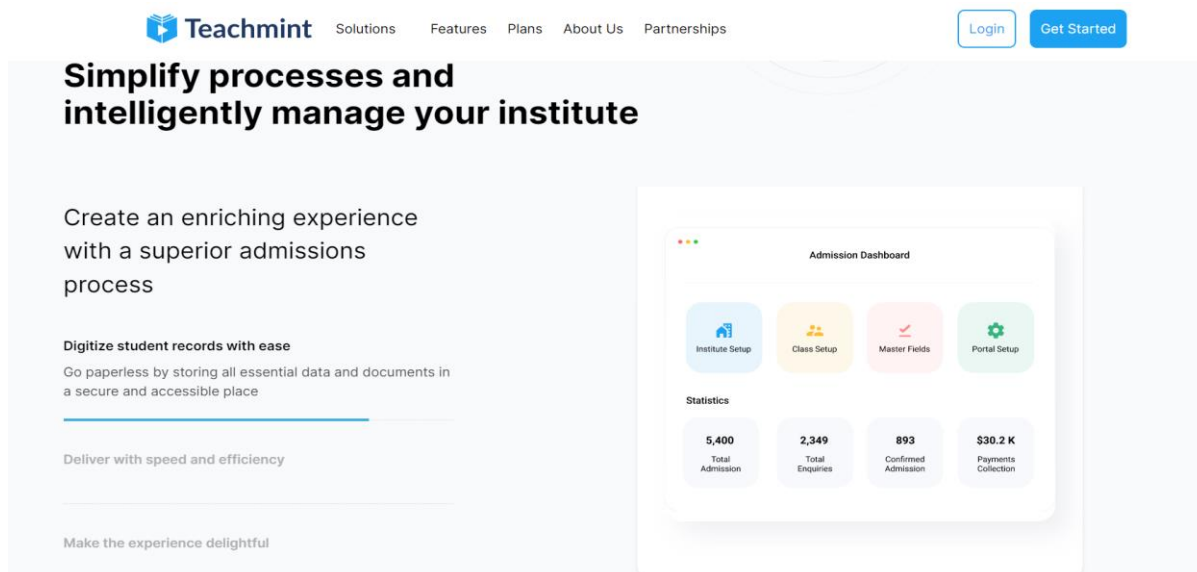


Рисунок 1.4 — Веб-сайт teachmint.com

Teachmint також забезпечує можливість відеоконференцій та спілкування вчителя з учнями. Рисунок 1.4 демонструє можливість вести заняття в режимі реального часу, задавати й отримувати відповіді на питання та спілкуватися віддалено. Крім того, платформа надає функціонал для моніторингу активності учнів і аналізу їх прогресу. Вчителі можуть відстежувати участь учнів, їх продуктивність та успішність, що допомагає оцінювати та покращувати процес навчання.

Окрім цього, Teachmint надає доступ до навчальних ресурсів, посібників та підтримки для вчителів. Вони можуть скористатися цими ресурсами для розширення своїх навичок у сфері викладання та використання платформи Teachmint.

Високі швидкості Інтернету та зростання віддалених форм навчання в останні роки призвели до зростання популярності платформ, подібних до Teachmint. Ці платформи стають все більш важливими для забезпечення продуктивного та ефективного навчання в умовах віддаленості, забезпечуючи вчителям і учням зручні інструменти для навчання та спілкування. Teachmint входить в цей ринок, надаючи вчителям і учням засоби для успішного

віддаленого навчання та управління класом.

Teachmint забезпечує можливість взаємодії вчителя і учнів через чат, віртуальну дошку, спільну роботу з документами та інші інструменти. Це створює інтерактивну навчальну атмосферу, де можна спілкуватися, ділитися думками та колективно працювати над завданнями як показано на рисунку 1.5.

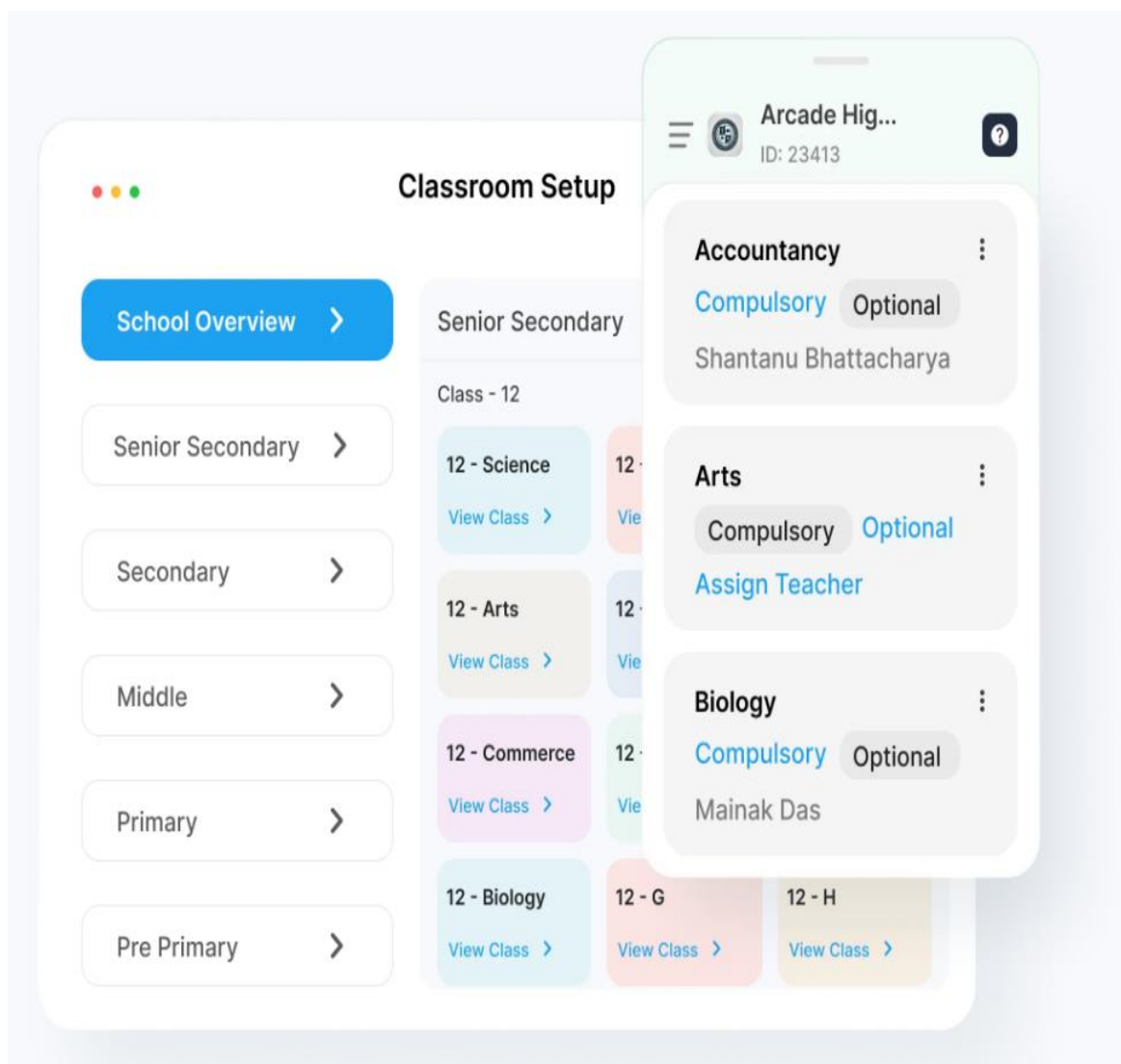


Рисунок 1.5 — Приклад створення класу на сайті teachmint.com

Також платформа надає інструменти для призначення та збору домашніх завдань, а також оцінювання успішності учнів. Вчителі можуть вести облік оцінок, встановлювати критерії оцінювання та надавати зворотний зв'язок стосовно робіт учнів.

## Висновок до розділу 1

В даному розділі було розглянуто перші адаптивні платформи для онлайн-навчання, наведено статистичні дані успішності платформи та запропоновано зміни.

Також було розглянуто платформу Moodle та Google Classroom з якими був реальний досвід роботи. При описі платформи Moodle було описані переваги та недоліки порівняно з розробленим додатком, також було наведено вигляд інтерфейсу цієї платформи. Також при аналізі Google Classroom було враховано переваги та недоліки порівняно з застосунком, наведено статистичні дані зростання компанії Google на період карантину, що свідчить про успішність цієї платформи, а також було підсумовано основні можливості Classroom.

Також було ознайомлено з застосунком який використовується як платформа для віддаленого навчання та управління класом. Перераховано переваги та наведено приклад створення нового класу.

Після аналізу існуючих аналогів, можна прийти до висновку, що розробка даного застосунку є виправданою, оскільки вона робить процес контролю набагато зручнішим та ефективнішим.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

## 2 АНАЛІЗ ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

### 2.1 Мова програмування Java

Мова Java була створена для інтерактивного телебачення, проте в ті часи ця мова була занадто передовою технологією.

Принципами створення Java-програмування були: "Простота, надійність, портативність, незалежність від платформи, безпека, висока продуктивність, багатопоточність, архітектурна нейтральність, об'єктно-орієнтованість, інтерпретованість і динамічність".

Після першого випуску Java до мови було додано багато додаткових можливостей. Зараз Java використовується в Windows-додатках, веб-додатках, корпоративних додатках, мобільних додатках, картках тощо. Кожна нова версія додає нові можливості в Java. [4-5]

Java - це потужна універсальна мова програмування. Її використовують для створення десктопних і мобільних додатків, обробки великих об'ємів даних, вбудованих систем тощо. За даними Oracle, компанії-власника Java, Java працює близько на 3 мільярдах пристроїв в світі, що робить її однією з найбільш популярних мов програмування. Оскільки Java є безоплатною та універсальною мовою, на ній створюють локалізовані та розповсюджені програми. Типові приклади використання Java:

- багато популярних мобільних, комп'ютерних та відеоігор створено на Java. Навіть сучасні ігри, в яких використовуються передові технології, такі як машинне навчання або віртуальна реальність;

- мову Java часто називають WORA (Write Once and Run Anywhere "Напиши один раз, запускай де завгодно"), що робить її ідеальною для децентралізованих хмарних додатків. Постачальники хмарних послуг обирають мову Java для запуску програм на широкому спектрі базових платформ;

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

–мова Java застосовується для механізмів обробки даних, які можуть працювати зі складними наборами даних і великими обсягами даних у режимі реального часу;

–Java - це джерело бібліотек машинного навчання. Завдяки своїй стабільності та швидкості мова стала вибором № 1 для розроблення додатків штучного інтелекту, таких як обробка природної мови та глибоке навчання;

–мова Java використовується для програмування датчиків і апаратного забезпечення периферійних пристроїв, які можуть незалежно підключатися до Інтернету.

Секрет популярності Java полягає в простоті її використання. Існують деякі причини, через які розробники віддають перевагу Java перед іншими мовами програмування.

Оскільки Java існує вже давно, для нових програмістів доступно безліч навчальних ресурсів. Детальна документація, друковані матеріали та курси допомагають розробникам протягом усього навчання. Крім того, новачки можуть почати писати код на Core Java, перш ніж переходити на Advanced Java.

Під час використання Java розробникам не потрібно щоразу писати нову функцію з нуля. Як альтернатива Java надає багату екосистему вбудованих функцій і бібліотек для розробки низки застосунків. Користувачі можуть завантажувати ненадійний код Java мережею і запускати його в безпечному середовищі, в якому він не може заподіяти жодної шкоди.

Ненадійний код не може заразити хост-систему вірусом, а також не можечитати або записувати файли з жорсткого диска. Рівні безпеки та обмеження в Java також легко налаштовуються.

Код Java може працювати на будь-якій базовій платформі, такій як Windows, Linux, iOS або Android, без перезапису. Користувачі можуть завантажувати ненадійний код Java мережею і запускати його в безпечному середовищі, в якому він не може заподіяти жодної шкоди. Ненадійний код не може заразити хост-систему вірусом.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

Коли мови програмування були вперше розроблені, вони поділялися на дві великі категорії залежно від того, як вони взаємодіяли з базовим обладнанням.

Компілятори. Уся програма написана в природному англійському синтаксисі за допомогою компільованих програм, а потім мова компілює (або перекладає) увесь код у машинний код. Після цього скомпільований код запускається на обладнанні.

Інтерпретатори. За допомогою інтерпретованих програм кожен високорівневий вираз коду інтерпретується в машинний код у режимі реального часу. Зафіксовані приписи негайно виконуються апаратним забезпеченням перед переглядом наступного припису.

Мова Java була першою мовою, яка об'єднала обидва описаних вище методи з використанням JVM. Компілятор коду Java називається віртуальною машиною Java. Будь-який файл Java спочатку компілюється в байт-код. Байт-код Java може працювати тільки в JVM. Потім JVM інтерпретує байт-код для запуску на базовій апаратній платформі.

Тому, якщо додаток працює на комп'ютері з Windows, JVM інтерпретує його для Windows. Однак під час роботи на платформі з відкритим вихідним кодом, такий як Linux, JVM інтерпретує його для Linux.

## 2.2 Використані фреймворки

Фреймворк – це готова модель в ІТ, заготовка, шаблон для програмної платформи, на основі якого можна дописати власний код. Це проміжний варіант за гнучкістю і складністю між написанням коду з нуля і використанням CMS. Якщо написання коду з нуля - це малювання будь-якими інструментами на чистому аркуші паперу, а використання CMS - розфарбовування, то використання framework - це малювання в блокноті в клітинку. З одного боку, ви все ще можете намалювати що завгодно і як завгодно. З іншого, вже є напрямні

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

лінії та межі блокнота, тож малювати простіше.

Фреймворки "беруть на себе" тисячі нюансів, наприклад роботу з файловою системою і базами даних, обробку помилок, захист програми.

Фреймворки підходять для створення як великих проєктів, так і простих сайтів і застосунків, які планується розвивати в майбутньому. Вони дають змогу правильно вибудувати бізнес-логіку. Здебільшого фреймворки використовуються для створення калькуляторів, інтернет-магазинів із нестандартним функціоналом, власних CRM, які не можна створити на CMS, десктопних і мобільних застосунків. На відміну від CMS, frameworks - це низькорівневе рішення, яке має більшу продуктивність і гнучкість. Дозволяє отримати готовий каркас для проєкту без втрати гнучкості в плані функціоналу.

### 2.2.1 Spring Boot

Spring - це програмне забезпечення з відкритим вихідним кодом, яке було створено для вирішення складності розробки корпоративних додатків. Він має багаторівневу архітектуру, яка допомагає розробникам вибирати компоненти, які можуть бути використані в подальшому, забезпечуючи при цьому цілісний фреймворк для процесу розробки додатків.

Spring framework є гнучким, оскільки підтримує та може бути інтегрований з різними технологіями:

- підтримує стиль REST веб-сервісів;
- підтримує управління транзакціями;
- розробник може взаємодіяти з різними базами даних;
- контейнери можуть вирішувати необхідні залежності, оскільки підтримують інтеграцію залежностей.

Spring Boot - це Java-фреймворк з відкритим вихідним кодом, який дозволяє створювати мікросервіси. Він розроблений командою Pivotal Team і призначений для створення готових до виробництва весняних додатків. У цій

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

частині ви дізнаєтеся про Spring Boot та його основні концепції.

Мікросервіс - це архітектура, що надає змогу створювати програми та розгортати сервіси незалежно. Кожен запущений сервіс використовує власний процес, і це дозволяє досягти легкої моделі підтримки бізнес-додатків.

Мікросервіси пропонують наступні переваги:

- мінімальну конфігурацію;
- легке розгортання;
- проста масштабованість;
- сумісність з контейнерами;
- зменшує час виробництва.

Spring Boot - це відмінна платформа для розробників на Java, яка дозволяє створювати готові до виробництва Spring-додатки з мінімальними зусиллями. Розробник може швидко почати працювати навіть з мінімальними налаштуваннями, не займаючись повним конфігуруванням Spring.

Фреймворк пропонує наступні переваги:

- значно збільшує продуктивність;
- легко розуміти і розробляти Spring-додатки;
- зменшує час розробки.
- даний фреймворк розроблений з наступними цілями:
- уникнути складних XML-конфігурацій у Spring;
- знайти більш простий спосіб спростити розробку готових до виробництва

Spring-додатків;

- скоротити час розробки та запустити застосунок незалежно;
- запропонувати простіший спосіб початку роботи з додатком.

Також, фреймворк обирають через наступні функції та переваги, які пропонуються:

- надає можливість гнучко налаштувати Java Beans, XML-конфігурації та бази даних для транзакцій;

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

- він має потужні функції для пакетної обробки та управління кінцевими точками REST;
- у фреймворкі всі налаштування відбуваються автоматично, не потрібно виконувати жодних ручних налаштувань;
- пропонується додаток на основі анотацій;
- полегшує управління залежностями;
- фреймворк має вбудований контейнер сервлетів.

Він здатний автоматично налаштувати ваш додаток на основі залежностей, які ви включаєте у проект за допомогою анотації `@EnableAutoConfiguration`. Наприклад, якщо в вашому проекті є залежність від бази даних MySQL, але ви не надали жодних конкретних налаштувань підключення до бази даних, то Spring Boot автоматично налаштує тестову базу даних у пам'яті для вас.

Точкою входу Spring Boot додатка є клас, що містить анотацію `@SpringBootApplication` та основний метод. У Spring Boot, за допомогою анотації `@ComponentScan`, відбувається автоматичне сканування всіх компонентів, які включені у ваш проект. [6]

### 2.2.2 Spring Web MVC

Архітектурний патерн MVC - це патерн проектування програмного забезпечення. Він виконує розділення роботи програми на наступні взаємопов'язані частини: модель, представлення і контролер. Модельна частина MVC охоплює дані додатку і, в більшості випадків, складається з POJO. Представлення (View) відповідає за відображення даних моделі і створює HTML-вивід, який може бути зрозумілий браузеру на стороні клієнта. Контролер відповідає за обробку запитів користувача і створення відповідної моделі, яка потім передається у View для відображення. Цей метод полегшує повторне використання коду та паралельну розробку. MVC популярний тому, що він

розділяє рівень логіки додатку і рівень користувацького інтерфейсу, що дозволяє розділити обов'язки.

Spring MVC – фреймворк мови Java, який призначений для написання веб-додатків. Фреймворк побудований на основі патерну MVC і володіє всіма основними функціями фреймворку, такими як Dependency Injection, Inversion of Control. Архітектурний дизайн Spring MVC можна використовувати для розробки гнучких застосунків. Він в основному розділяє різні аспекти програми, такі як вхідна логіка, логіка інтерфейсу користувача та бізнес-логіка.

MVC в основному використовуються для полегшення тестування, а також для створення зручної для SEO маршрутизації URL-адрес. Він також використовується для інтеграції розширень у фреймворк.

Структурну схему застосунку зображено на креслинику ІА94.190БАК.003 ДЗ. На цій схемі можна побачити архітектуру Spring MVC, зв'язки компонентів системи, та обробку запитів за допомогою HTTP.

Фреймворк Spring Web model-view-controller (MVC) розроблений навколо DispatcherServlet, який обробляє всі HTTP-запити та відповіді. Робочий процес обробки запитів Spring Web MVC DispatcherServlet проілюстровано на рисунку 2.1. [7]

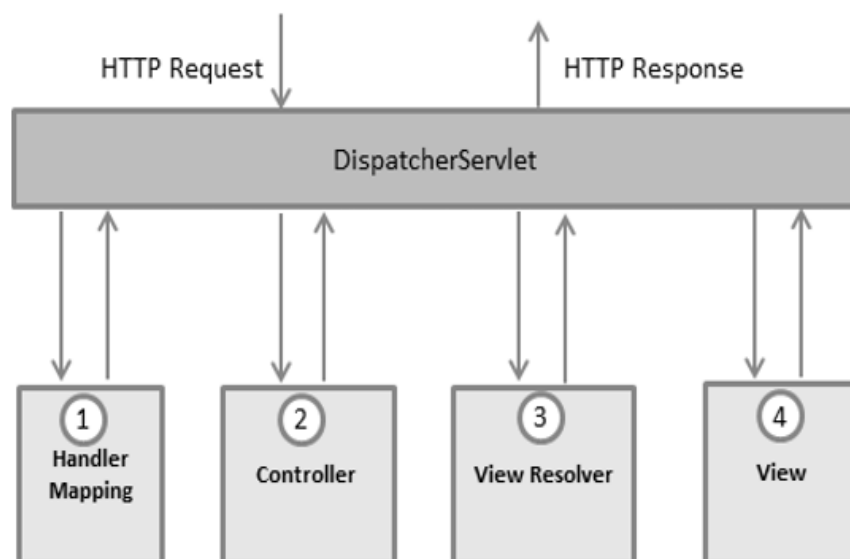


Рисунок 2.1 – Робочий процес обробки запитів Spring Web MVC DispatcherServlet

Послідовність подій, що відповідають вхідному HTTP-запиту до DispatcherServlet:

–після отримання HTTP-запиту DispatcherServlet звертається до HandlerMapping для виклику відповідного контролера;

–контролер приймає запит і викликає відповідні сервісні методи на основі використаного методу GET або POST. Сервісний метод встановлює дані моделі на основі визначеної бізнес-логіки і повертає ім'я подання до DispatcherServlet;

–сервлет DispatcherServlet отримає допомогу від ViewResolver, щоб підібрати визначене подання для запиту;

–після того, як представлення завершено, DispatcherServlet передає дані моделі до представлення, яке остаточно відображається у браузері.

Всі вищезгадані компоненти, тобто HandlerMapping, Controller і ViewResolver є частинами WebApplicationContext, який є розширенням plainApplicationContext з деякими додатковими функціями, необхідними для веб-додатків.

DispatcherServlet - це фронт-контролер і стоїть наряду з найважливішими компонентами веб-фреймворку Spring MVC. Фронт-контролер - це типова структура у веб-додатках, яка отримує запити та делегує їх обробку іншим компонентам додатку.

DispatcherServlet діє як єдина точка входу для клієнтських запитів до веб-додатку Spring MVC, перенаправляючи їх до відповідних контролерів Spring MVC для обробки. DispatcherServlet є переднім контролером, який також допомагає з роздільною здатністю подання, обробкою помилок, роздільною здатністю локалі, роздільною здатністю теми та іншими речами.

DispatcherServlet повинен бути вказаний в дескрипторі розгортання або файлі web.xml, як і будь-який інший клас сервлетів. За допомогою web.xml конфігурації можна вказати додаткові перевірки для точки входу від клієнтської частини та перевірити вхідні дані без виклику сервісів, що використовує контролер.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

Можна виділити наступні переваги Spring MVC.

Різні ролі: Spring MVC має окремі ролі, які можуть бути виконані за допомогою спеціалізованого об'єкта. До таких ролей відносяться командний об'єкт, об'єкт моделі, контролер, диспетчерський сервлет, вирішувач представлень, валідатор та інші.

Легкість: Контейнер, що використовується для розробки та розгортання додатків, використовує легкий сервлет.

Швидка розробка: Пружинний фреймворк MVC забезпечує швидку та паралельну розробку. Це допомагає розробнику завершити проект вчасно.

Надійна та потужна конфігурація: Він забезпечує потужну конфігурацію для класів додатків та фреймворку. Ці конфігурації включають легкі прості посилання, такі як бізнес-об'єкти та веб-контролери.

Бізнес-код: Надає багаторазові бізнес-коди, які дозволяють розробнику використовувати існуючі бізнес-об'єкти замість створення нових об'єктів.

Легке тестування: Spring зазвичай використовує Java Beans, що дозволяє розробнику вставляти дані за допомогою простих методів.

Відображення: Забезпечує гнучке відображення, що дозволяє легко перенаправляти сторінку.

Ін'єкція залежностей: Інверсія Control або ін'єкція залежностей дозволяє розробнику не створювати повне середовище для залежностей.

Spring MVC забезпечує чіткий розподіл обов'язків, що дозволяє нам швидко розробляти та ретельно аналізувати наші додатки. Сервлет диспетчера, вирішувачі представлень, контролери, ModelAndView, представлення, моделі та атрибути сеансів - все це абсолютно різні елементи, які відповідають виключно за один аспект. Як результат, MVC надає нам велику свободу. Він побудований на інтерфейсах із заданими класами реалізації, і ми можемо використовувати кастомні інтерфейси для налаштування кожного аспекту фреймворку. Ще одним ключовим моментом є те, що користувачі не обмежені технологією єдиного представлення, а навпаки, мають можливість вибирати ті, які потрібні. [8]

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

### 2.2.3 Spring Security

Spring Security - це фреймворк, який дозволяє програмісту накладати обмеження безпеки на веб-додатки на основі Spring-фреймворку за допомогою JEE-компонентів. Коротше кажучи, це бібліотека, яку можна використовувати, розширювати та налаштовувати відповідно до потреб програміста. Оскільки вона є членом тієї ж родини Spring, вона чудово поєднується з Spring Web MVC. Її основною сферою застосування є обробка автентифікації та авторизації на рівні веб-запитів, а також на рівні виклику методів. Можливо. Найбільша перевага цього фреймворку полягає в тому, що він є потужним, але при цьому дуже добре налаштовується у своїй реалізації. Хоча він слідує конвенції Spring щодо конфігурації, програмісти можуть вибрати між стандартними положеннями та налаштуванням відповідно до своїх потреб.

Автентифікація означає, що, отримуючи доступ до певних ресурсів з обмеженим доступом, користувач дійсно є тією особою, яка має на це право. Існує два процеси, щоб переконатися, що користувач є автентичним: ідентифікація та верифікація. Наприклад, користувач ідентифікується за допомогою імені користувача та пароля, які зазвичай зберігаються в базі даних; LDAP (Lightweight Directory Access Protocol, полегшений протокол доступу до служб каталогів); або CAS (Central Authentication Service, протокол єдиного входу в мережу Інтернет).

Авторизація визначає ступінь права користувача на доступ до обмежених ресурсів. Вона гарантує, що користувачеві дозволено доступ лише до тих частин ресурсу, які він в змозі використовувати.

Для прикладу, адміністратор має невичерпний доступ до властивостей програми і може змінювати або проводити маніпуляції над ними - на краще чи на навпаки. Проте, звичайний користувач або користувач гість має більш контрольований доступ і не використовує ті ж самі права, що й користувач адміністратор. Такий процес є розподіленням ролей користувача. У всіх веб-додатку це робиться за допомогою захисту на

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

основі URL-адрес. Фреймворк надає фільтри для забезпечення ролі захисту програми.[9]

Проте захист на основі URL-адрес не є являється практичним механізмом і часто може бути використаний не за його призначенням. Зловмисники можуть маніпулювати URL-адресою і отримати доступ до методу, що насправді розрахований для адміністративного користувача. Оскільки в системі на основі URL-адрес виклики методів з обмеженим доступом надсилаються за допомогою гіперпосилань, досить легко повторно створити той самий виклик методу з URL-адреси і надіслати його на сервер. Сервер може наївно виконати обмежені операції, не перевіривши роль користувача, який викликав запит. Тому, щоб вирішити цю проблему, Spring пропонує безпеку на рівні методів. Це означає, що лише певні авторизовані користувачі можуть викликати обмежені методи, і просте повторне створення URL-адреси та надсилання її на сервер не призведе до їх виконання.

Spring Security має наступні переваги Spring Security:

- інтеграція з API сервлетів;
- розширювана підтримка аутентифікації та авторизації;
- захист від таких атак, як фіксація сеансу, підміна кліків;
- інтеграція з Spring MVC;
- можливість захистити додаток від атак грубої сили;
- переносимість;
- захист від CSRF-атак;
- підтримка конфігурації Java. [10]

У Spring Security модуль безпеки складається з окремих jar-файлів залежно від його функціональності. Основне призначення - дозволити користувачеві інтегруватися відповідно до вимог.

Переваги Spring Security не обмежуються допомогою з автентифікацією та авторизацією. Він також може допомогти в застосуванні найкращих практик для збереження користувачів, а саме, створити функцію реєстрації.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

У процесі створення функції реєстрації є кілька поширених помилок, які можуть призвести до серйозних проблем з безпекою.

Одна з поширених помилок - це збереження паролів користувачів у відкритому вигляді без будь-якого захисту (хешування). Це не лише дає змогу хакерам, які можуть отримати доступ до вашого сервера, переглядати паролі безпосередньо, але й наражає користувачів на загрозу злому інших акаунтів, оскільки більшість людей використовують однакові паролі для всіх своїх облікових записів.

Іншою поширеною помилкою є спроба винайти велосипед і створити власний алгоритм шифрування/хешування замість того, щоб використовувати відомі, перевірені, перевірені алгоритми.

Spring Security дозволяє нам призначити безпечний шифратор паролів для нашого об'єкта UserDetails, щоб запобігти цим помилкам. За замовчуванням він використовує BCrypt для шифрування паролів, який вважається добре відпрацьованим алгоритмом для кодування паролів. Також можна встановити кількість раундів хешування (або стійкість, як впливає з назви параметра) і безпечну реалізацію випадкового алгоритму, яка буде використовуватися в процесі.

У застосунках, де back end і front end дуже пов'язані між собою, наприклад, у Spring MVC-додатках або будь-яких інших MVC-фреймворках: Кожного разу, коли користувач входить в систему, на сервері зберігається сесія, яка містить інформацію про цього користувача. Такі операції, як отримання, кешування, знищення цієї сесії називаються управлінням сесіями. Spring Security надає механізми для управління цим об'єктом сеансу. Він створить сесію, коли користувач увійде в систему, знищить її, коли він вийде з системи, і дозволить нам встановити значення таймауту.

Spring Security також вживає додаткових заходів, щоб переконатися, що сеанси використовуються безпечно:

–він може бути налаштований на відключення перезапису URL-адрес;

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

–він автоматично переносить сесію, коли користувач знову входить в систему, щоб уникнути фіксації сесії;

–він дозволяє нам використовувати прапори httpOnly та безпеки для сесійних файлів cookie, щоб захистити наші файли cookie. [11]

На рисунку 2.2 можна ознайомитись з принципом роботи архітектури автентифікації Spring Security.

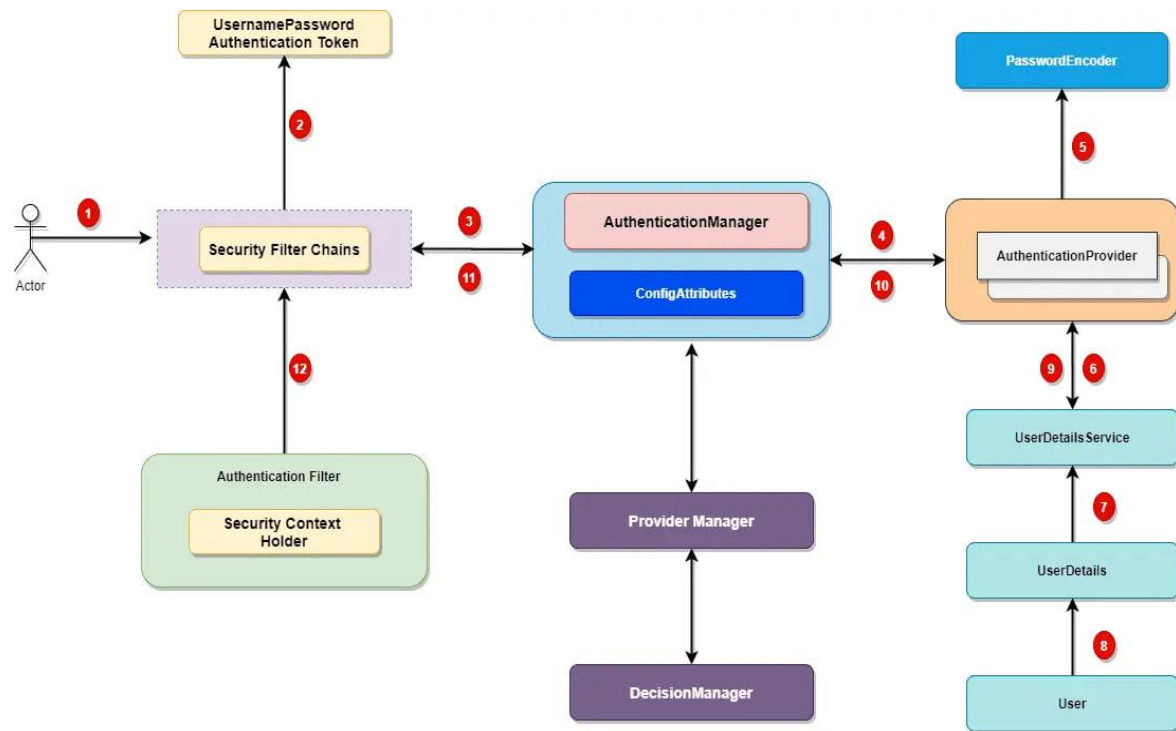


Рисунок 2.2 – Архітектура Spring Security

## 2.2.4 Hibernate

Hibernate - це інструмент об'єктно-реляційного відбиття (ORM) який має код з відкритим доступом, він дає фреймворк для відображення об'єктно-орієнтованих моделей доменів у реляційні бази даних для веб-додатків. Об'єктно-реляційне відтворення базується на контейнеризації об'єктів та абстракції, яка забезпечує цю можливість. Абстракція дозволяє адресувати об'єкти, отримувати до них доступ і маніпулювати ними без необхідності враховувати, як вони пов'язані з їхніми джерелами даних.

Ніibernate ORM керує відображенням класів Java до таблиць бази даних і типів даних Java до типів SQL даних, а також забезпечує запити та пошук.

Всі зміни відбуваються в самому джерелі даних, тому коли джерела або їхні API змінюються, додатки, які використовують ORM, не потребують внесення змін або навіть свідомості про ці зміни. Аналогічно, програмісти можуть мати узгоджене уявлення про об'єкти з плином часу, хоча джерела, які їх надають, приймачі, які їх отримують, і додатки, які отримують до них доступ, можуть змінюватися. [12]

Ніibernate є вільно доступним для завантаження і поширюється на умовах ліцензії з відкритим вихідним кодом GNU Lesser General Public License (LGPL).

Ніibernate - це об'єктно-реляційна служба збереження даних і запитів з відкритим вихідним кодом для будь-яких Java-додатків. Ніibernate зіставляє класи Java з таблицями бази даних і типи даних Java з типами даних SQL, звільняючи розробника від найпоширеніших завдань програмування, пов'язаних зі збереженням даних.

Ніibernate знаходиться між традиційними об'єктами Java та сервером бази даних і виконує всю роботу по збереженню цих об'єктів на основі відповідних механізмів та шаблонів O/R, схему яких зображено на рисунку 2.3.

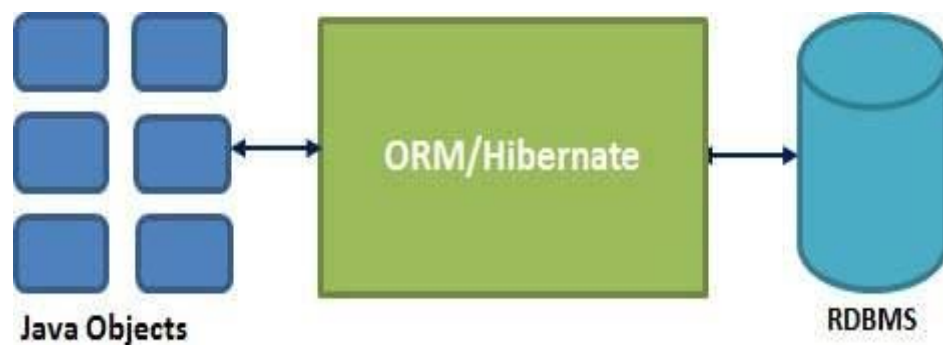


Рисунок 2.3 – Розміщення Ніibernate між Java та БД

Можна виділити наступні переваги:

–Ніibernate піклується про зіставлення класів з таблицями бази даних за допомогою XML-файлів і без написання жодного рядка коду;

–надає прості API для зберігання та отримання об'єктів Java безпосередньо в базу даних та з неї;

–якщо в базі даних або в будь-якій таблиці відбуваються зміни, то вам потрібно змінити лише властивості XML-файлу;

–абстрагує незнайомі типи SQL і надає можливість працювати зі знайомими об'єктами Java;

–для роботи Hibernate не потрібен сервер додатків;

–маніпулює складними асоціаціями об'єктів вашої бази даних;

–мінімізує доступ до бази даних за допомогою стратегій вибірки;

–забезпечує простий запит даних.

Hibernate підтримує майже всі основні СУБД. Нижче наведено список деяких механізмів баз даних, які підтримуються Hibernate:

–HSQL Database Engine;

–DB2/NT;

–MySQL;

–PostgreSQL;

–FrontBase;

–Oracle;

–Microsoft SQL Server Database;

–Sybase SQL Server;

–Informix Dynamic Server.

### 2.3 MySQL

MySQL – найпопулярніша у світі база даних з відкритим вихідним кодом. За даними DB-Engines, MySQL посідає друге місце за популярністю після Oracle Database. MySQL забезпечує роботу багатьох найпопулярніших додатків, включаючи Shopify, Netflix, Airbnb, Twitter, Uber, Facebook, та Booking.com.

БД – є основним сховищем даних для всіх програмних застосунків. Наприклад, щоразу, коли хтось здійснює веб-пошук, входить в обліковий запис або завершує транзакцію, система баз даних зберігає інформацію, щоб до неї можна було отримати доступ у майбутньому.

Реляційна база даних зберігає дані в окремих таблицях, а не в одному великому сховищі. Структура бази даних організована у фізичні файли, оптимізовані для швидкості роботи. Логічна модель даних з такими об'єктами, як таблиці даних, подання, рядки та стовпці, пропонує гнучке середовище програмування. Ви встановлюєте правила, що регулюють зв'язки між різними полями даних, наприклад, один до одного, один до багатьох, унікальні, обов'язкові чи необов'язкові, а також "показчики" між різними таблицями. База даних забезпечує дотримання цих правил, тому завдяки добре спроектованій базі даних програма ніколи не побачить неузгоджених, дубльованих, порожніх, застарілих або відсутніх даних.

MySQL – це клієнт-серверна система, що містить в собі багатопотокові SQL-сервера, що підтримує різні інтерфейси, декількох бібліотек і клієнтських програм, інструментів адміністрування та широкого спектру API. Також MySQL надається у вигляді вбудованої багатопотокової бібліотеки, яку є змога підключити до свого додатку, щоб отримати менший, швидший і простіший в управлінні автономний продукт.

MySQL – швидка, надійна, масштабована і проста у використанні. Спочатку її було розроблено для швидкої обробки великих баз даних, і вона вже багато років використовується у виробничих середовищах з високими вимогами до продуктивності.

Хоча MySQL постійно розвивається, вона пропонує багатий і корисний набір функцій. Зв'язність, швидкість і безпека MySQL роблять її дуже придатною для доступу до баз даних в Інтернеті.

Основні переваги MySQL включають:

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

–розробники можуть встановити MySQL за лічені хвилини, а базою даних легко керувати.

–MySQL є однією з найбільш зрілих і широко використовуваних баз даних. Вона була протестована в найрізноманітніших сценаріях протягом більше 25 років, в тому числі багатьма найбільшими світовими компаніями. Організації покладаються на MySQL для запуску критично важливих для бізнесу додатків через її надійність.

**Масштабованість:** MySQL масштабується, щоб задовольнити вимоги найбільш популярних додатків. Власна архітектура реплікації MySQL дозволяє таким організаціям, як Facebook, масштабувати додатки для підтримки мільярдів користувачів.

**Продуктивність:** MySQL HeatWave працює швидше і дешевше, ніж інші сервіси баз даних, що підтверджується численними стандартними галузевими тестами, включаючи TPC-H, TPC-DS і CN-benCHmark.

**Висока доступність:** Надається повний набір власних, повністю інтегрованих технологій реплікації для забезпечення високої доступності та аварійного відновлення. Для критично важливих для бізнесу додатків і для виконання зобов'язань за угодою про рівень обслуговування клієнти можуть досягти.

–Цільова точка відновлення = 0 (нульова втрата даних)

–Цільовий час відновлення = секунди (автоматичне обхід відмови)

**Безпека:** Безпека даних передбачає захист і відповідність галузевим і державним нормам, включаючи Загальний регламент захисту даних Європейського Союзу, Стандарт безпеки даних індустрії платіжних карток, Закон про переносимість і підзвітність медичного страхування, а також Керівництва з технічної реалізації безпеки Агентства оборонних інформаційних систем (Defense Information Systems Agency). MySQL Enterprise Edition пропонує розширені функції безпеки, містячи в собі автентифікацію/авторизацію, прозоре шифрування даних, аудит, маскування даних і брандмауер бази даних.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

Гнучкість: MySQL Document Store надає користувачам максимальну гнучкість у розробці традиційних додатків баз даних SQL і NoSQL без схем. Розробники можуть змішувати і поєднувати реляційні дані і JSON-документи в одній базі даних і додатку.

MySQL має різні варіанти використання, наприклад:

Хмарні додатки: MySQL дуже популярна в хмарі. MySQL HeatWave - це повністю керована служба баз даних, що працює на основі інтегрованого прискорювача запитів в пам'яті HeatWave. Це єдиний хмарний сервіс баз даних, який поєднує транзакції, аналітику та послуги машинного навчання (ML) в одній базі даних MySQL, забезпечуючи безпечну аналітику в режимі реального часу без складності, затримок та витрат, пов'язаних з дублюванням ETL. MySQL HeatWave в 6,5 разів швидша за Amazon Redshift за половину вартості, в 7 разів швидша за Snowflake за п'яту частину вартості та в 1 400 разів швидша за Amazon Aurora за половину вартості. З MySQL HeatWave ML розробники та аналітики даних можуть створювати, тренувати та пояснювати моделі машинного навчання в повністю автоматизований спосіб - у 25 разів швидше, ніж Amazon Redshift ML за 1% від вартості.

Електронна комерція: Багато з найбільших у світі додатків для електронної комерції (наприклад, Shopify, Uber і Booking.com) запускають свої транзакційні системи на MySQL. Це популярний вибір для управління профілями користувачів, обліковими даними, користувацьким контентом, фінансовими даними, включаючи платежі, та виявлення шахрайства.

Соціальні платформи: Facebook, Twitter та LinkedIn є одними з найбільших соціальних мереж у світі, які покладаються на MySQL.

Управління контентом: На відміну від одноцільових баз даних документів, MySQL дозволяє використовувати як SQL, так і NoSQL в одній базі даних. Сховище документів MySQL дозволяє виконувати CRUD-операції і використовувати можливості SQL для запитів даних з JSON-документів для звітності та аналітики.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

SaaS та ISV: Понад 2 000 ISV, OEM-виробників і VAR-виробників, включаючи Ericsson, F5 і IBM, покладаються на MySQL як на вбудовану базу даних, щоб зробити свої додатки, обладнання та прилади більш конкурентоспроможними, швидше вивести їх на ринок і знизити собівартість проданих товарів. MySQL також є базою даних для популярних SaaS-додатків, зокрема Zendesk та HubSpot.

Серед інших популярних додатків, що використовують MySQL, - онлайн-ігри, цифровий маркетинг, системи роздрібної торгівлі та системи моніторингу Інтернету речей.

Локальні додатки з MySQL Enterprise Edition: MySQL Enterprise Edition включає найповніший набір розширених функцій, інструментів управління та технічної підтримки для досягнення найвищих рівнів масштабованості, безпеки, надійності та безвідмовності MySQL. Вона зменшує ризик, вартість і складність розробки, розгортання та управління критично важливими для бізнесу додатками MySQL. Він надає функції безпеки, включаючи резервне копіювання, моніторинг, брандмауер, аудит, прозоре шифрування даних і автентифікацію MySQL Enterprise, щоб допомогти клієнтам захистити дані і досягти відповідності нормативним і галузевим вимогам. [13]

## 2.4 Swagger

Swagger – це набір інструментів з відкритим вихідним кодом для написання API на основі REST. Він значно спрощує процес написання API, визначаючи стандарти та надаючи інструменти, необхідні для написання надійних, безпечних, продуктивних та масштабованих API.

У сучасному світі програмного забезпечення майже не залишилося систем, що працюють в Інтернеті, які б не використовували API. Весь дизайн архітектури мікросервісів базується на API.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

Основні інструменти Swagger включають:

–Swagger Editor - браузерний редактор, за допомогою якого ви можете писати визначення OpenAPI;

–Swagger UI - перетворює визначення OpenAPI на інтерактивну документацію;

–Swagger Codegen - генерує серверні заглушки та клієнтські бібліотеки з визначення OpenAPI;

–Swagger Editor Next (бета-версія) - браузерний редактор, за допомогою якого ви можете писати та переглядати визначення OpenAPI та AsyncAPI;

–Swagger Core - бібліотеки, пов'язані з Java, для створення, споживання та роботи з визначеннями OpenAPI;

–Swagger Parser - окрема бібліотека для розбору визначень OpenAPI;

–Swagger APIDom - надає єдину уніфіковану структуру для опису API на різних мовах опису та форматах серіалізації. [14]

Swagger має два підходи до написання документації:

Документація пишеться на підставі коду.

–цей підхід позиціонується як "дуже просто". Нам достатньо додати кілька залежностей у проєкт, додати конфігурацію і вже ми матимемо потрібну документацію, хоч і не настільки описану, як ми хотіли;

–код проєкту стає не дуже читабельним від великої кількості анотацій і опису в них;

–уся документація буде вписана в нашому коді (всі контролери і моделі перетворюються на такий собі Java Swagger Code);

–підхід не радять використовувати, якщо є можливості, але його дуже просто інтегрувати.

Документація пишеться окремо від коду.

–цей підхід вимагає знати синтаксис Swagger Specification;

–документація пишеться або в YAML/JSON файлі

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

## 2.5 JavaScript

JavaScript – це мова програмування, яка є динамічною. Вона має простий синтаксис і часто використовується для створення веб-сторінок, де вона взаємодіє з користувачем і додає динамічні ефекти. Це інтерпретованою з об'єктно-орієнтованими можливостями.

Спочатку JavaScript була відома як LiveScript, але компанія Netscape змінила її назву на JavaScript, можливо, через ажіотаж, який викликала мова Java. Вперше JavaScript з'явився в Netscape 2.0 у 1995 році під назвою LiveScript. Ядро мови загального призначення було вбудовано в Netscape, Internet Explorer та інші веб-браузери.

JavaScript – найбільш поширена мова програмування на стороні клієнта. Скрипт повинен бути включений в HTML-документ або посилатися на нього, щоб код міг бути інтерпретований браузером.

Це означає, що веб-сторінка може бути більш ніж просто статичним HTML. Вона може містити програми на JavaScript, які взаємодіють з користувачем, керують браузером і динамічно генерують HTML-контент.

Механізм JavaScript на стороні клієнта надає багато переваг над традиційними сценаріями CGI на стороні сервера. Наприклад, розробник може застосовувати JavaScript для перевірки, чи введена користувачем адреса електронної пошти є дійсною в полі форми.

Код JavaScript активується, коли користувач відправляє форму, і тільки у випадку, якщо всі введені дані є дійсними, вони будуть передані на веб-сервер.

JavaScript може використовуватися для відстеження подій, ініційованих користувачем, таких як натискання кнопок, навігація по посиланнях та інші дії, які користувач ініціює явно чи неявно.

Перевагами використання JavaScript:

–для користування достатньо лише браузера;

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

–менше обміну даними з сервером - можна перевірити введені користувачем дані перед відправкою сторінки на сервер. Це дозволяє заощадити серверний трафік, зменшуючи навантаження на сервер;

–відразу отримувати зворотний зв'язок від відвідувачів – користувачам не треба очікувати перезавантаження веб-сторінки, щоб перевірити, чи не помилились вони при вводі;

–JavaScript дозволяє створювати інтерфейси з підвищеною інтерактивністю, що означає, що їх реакція залежить від дій користувача, таких як наведення курсора миші на елементи або активація їх за допомогою клавіатури;

–багатші інтерфейси з можливістю використовувати JavaScript для включення таких елементів, як компоненти перетягування і повзунки, щоб надати відвідувачам вашого сайту багатий інтерфейс.

## Висновок до розділу 2

В даному розділі було описано використанні технології, як використовуються для розробки застосунку за темою дипломної роботи.

Насамперед, було обрано і описано мову програмування, вибір впав на мову Java, ця мова ідеально підходить для вирішення даної задачі. Було описано переваги даної мови, для яких цілей її краще всього застосовувати.

Також, було описано фреймворки, його поняття, а також перелічені використані фреймворки для розробки застосунку, такі як: Spring Boot, Spring MVC, Spring Security, Hibernate. В розділі детально описано для яких цілей використовується той чи інший фреймворк. Перелічено їх переваги, та наведено принцип їх роботи. Було описано обрану базу даних, в якій безпосередньо будуть зберігатись та чи інша інформація, наприклад, логіни та паролі, оцінки тощо. Вибір впав на таку базу даних, як MySQL, в розділі перелічені цієї БД.

Отже, в розділі було проаналізовано всі технології які використовуються в практичній частині дипломної роботи.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

## 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1 Структура проєкту

Як вже було зазначено, для реалізації додатку було обрано мову програмування Java. Використанні фреймворки наступні: Spring Boot, Spring MVC, Spring Security та Hibernate, вище було зазначено які функції вони виконують та для яких цілей їх було обрано. На рисунку 3.1 представлена структура проєкту, вміст папок та їх назви.

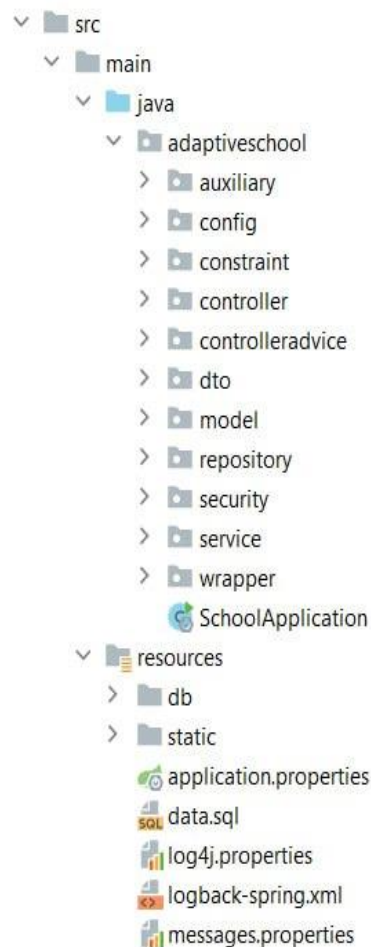


Рисунок 3.1 – Структура проєкту

config – зберігаються конфігураційні файли та налаштування, які використовуються для налаштування програми або різних компонентів проєкту. Основна функція цієї папки полягає в тому, щоб зберегти параметри та налаштування програми окремо від вихідного коду, що дозволяє їх легко змінювати без необхідності перекомпіляції коду.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

controller – зберігаються класи, які відповідають за обробку вхідних запитів та керування потоком виконання програми. Основна функція цієї папки полягає в тому, щоб відокремити логіку обробки запитів від моделей даних та відображення результатів.

controlleradvice – зберігаються класи, які відповідають за обробку помилок та глобальне налаштування для контролерів. Основна функція цієї папки полягає в тому, щоб централізовано керувати помилками та виконувати спільні операції для контролерів усього проекту.

dto – зберігаються класи, які представляють об'єкти для передачі даних між різними компонентами системи. Основна функція цієї папки полягає в тому, щоб відокремити моделі даних від їх представлення та спростити обмін даними між різними частинами програми.

model – зберігаються класи, які відповідають за модель даних або логіку програми. Основна функція цієї папки полягає в тому, щоб організувати структуру проекту та розділити різні компоненти програми для кращої читабельності та керованості.

repository – зберігаються інтерфейси та класи, які відповідають за доступ до даних та взаємодію з базою даних або іншими джерелами даних. Основна функція цієї папки полягає в тому, щоб відокремити логіку доступу до даних від решти програми та забезпечити структуроване та просте використання даних.

security – зберігаються класи та конфігурації, які відповідають за забезпечення безпеки додатка. Основна функція цієї папки полягає в тому, щоб захистити ресурси, обмежити доступ і контролювати аутентифікацію та авторизацію користувачів.

service – зберігаються класи, які містять бізнес-логіку та виконують певні операції над даними. Основна функція цієї папки полягає в тому, щоб відокремити логіку бізнес-процесів від контролерів та репозиторіїв, забезпечуючи централізований доступ до функцій, необхідних для обробки даних.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

На креслинику ІА94.190БАК.003 Д1 наведено діаграму класів, зв'язок між ними, присутність класів, інтерфейсів та їх зв'язків. Важливо відзначити, що в цій діаграмі також можуть бути елементи, які описують поведінку, але їх використання є характерним для інших типів діаграм. Для об'єктно-орієнтованого програмування цей тип діаграми є найкращим варіантом, оскільки він охоплює всі використовувані парадигми.

На діаграмі зображено різні типи зв'язків між класами, такі як реалізація, асоціація, агрегація та успадкування. Асоціація вказує на те, що один екземпляр класу пов'язаний з іншим екземпляром класу. Асоціація може бути бінарною, якщо вона відбувається лише між двома класами, або n-арною, якщо вона відбувається між багатьма класами.

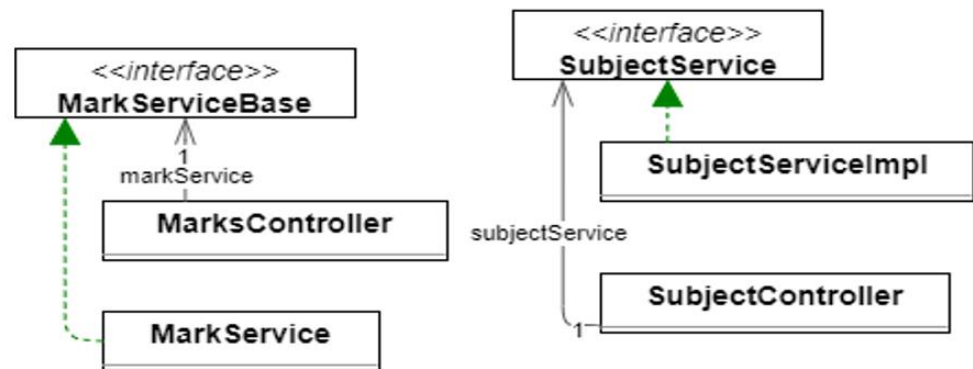


Рисунок 3.2 – Залежність між інтерфейсами та класами

Залежність між класами та інтерфейсами зображено на рисунку 3.2. Зміни в одному елементі можуть призводити до змін в іншому елементі. Наприклад, зміни в реалізації сервісу можуть впливати на поведінку контролера.

Наслідування використовується, коли клас успадковує поведінку абстрактного класу контролера. Це можна описати як часткове узагальнення, де підтип є частковою формою над-типу.

Агрегація застосовується, коли потрібно відношення між цілим та його частинами. Агрегація виникає, коли клас є контейнером для інших класів.

Час існування агрегованих класів не залежить від часу існування класу-контейнера. Це також називається агрегацією по посиланню.

Композиція є більш суворою формою агрегації. Класи, що проходять композицію, пов'язані з класом-контейнером за часом існування. Якщо клас-контейнер буде знищений, то знищаться всі класи, які були вкладені в нього через композицію.

### 3.2 Автентифікація

Автентифікація - це процес визначення того, чи є хтось або щось насправді тим, за кого себе видає. Технологія автентифікації забезпечує контроль доступу до систем, перевіряючи, чи відповідають облікові дані користувача обліковим даним у базі даних авторизованих користувачів або на сервері автентифікації даних. Таким чином, автентифікація забезпечує безпеку систем, безпеку процесів та інформаційну безпеку підприємства.

Існує кілька типів автентифікації. З метою ідентифікації користувача, користувачі зазвичай ідентифікуються за допомогою ідентифікатора користувача, а автентифікація відбувається, коли користувач надає облікові дані, такі як пароль, який відповідає його ідентифікатору користувача. Практика вимагати ідентифікатор користувача та пароль відома як однофакторна автентифікація (SFA). В останні роки компанії посилюють автентифікацію, запитуючи додаткові фактори автентифікації, такі як унікальний код, який надається користувачеві через мобільний пристрій при спробі входу у систему.

За вхід у систему та перевірку логіна та пароля відповідає `AuthenticationController`. Коли користувач переходить за посиланням `localhost:8080/singin`, то буде виклик контролера а саме метода `createAuthenticationToken`, що зображено на рисунку 3.3, до якого буде передано дані користувача. Контролер для перевірки даних викликає інтерфейс `AuthenticationManager` з отриманими даними від користувача та перевіряє

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

чи були передані логін та пароль для автентифікації у системі, потім буде згенеровано JWT токен, що буде збережено на клієнтській частині.

```
@PostMapping("/signin")
public ResponseEntity<> createAuthenticationToken(
    @ApiParam(value = "Login and Password", required = true)
    @RequestBody JwtAuthenticationRequest authenticationRequest
) {
    authenticate(authenticationRequest.getUsername(), authenticationRequest.getPassword());

    final JwtUser userDetails = (JwtUser) userDetailsService.loadUserByUsername(authenticationRequest.getUsername());
    final String token = jwtTokenUtil.generateToken(userDetails);
    HttpHeaders headers = new HttpHeaders();
    headers.add(tokenHeader, headerValue: tokenPrefix + token);
    logger.info("User {} successfully authenticated", authenticationRequest.getUsername());

    return new ResponseEntity(headers, HttpStatus.NO_CONTENT);
}
```

Рисунок 3.3 – Фрагмент коду з автентифікацією

JWT токен генерується за допомогою окремого класу JwtTokenUtil. Для майбутніх запитів буде використано токен, що користувач отримав при вході у систему, перед кожним запитом буде перевірено, що сесія користувача ще активна. Генерація токена використовує шифрування, що складається з алгоритму шифрування, секретного ключа та часу дії токена. Ці дані записано у файлах застосунку а серверна частина отримує їх за допомогою гетерів, які зображені на рисунку 3.4.

```
@Value("${jwt.secret}")
private String secret;

@Value("${jwt.expiration}")
private Long expiration;

@Value("${jwt.globalExpiration}")
private Long globalExpiration;

public String getUsernameFromToken(String token) {
    return getClaimFromToken(token, Claims::getSubject);
}

public Date getIssuedAtDateFromToken(String token) {
    return getClaimFromToken(token, Claims::getIssuedAt);
}

public Date getExpirationDateFromToken(String token) {
    return getClaimFromToken(token, Claims::getExpiration);
}

public <T> T getClaimFromToken(String token, Function<Claims, T> claimsResolver) {
    final Claims claims = getAllClaimsFromToken(token);
    return claimsResolver.apply(claims);
}
```

Рисунок 3.4 – Генерація JWT токена

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

Якщо сесія користувача вичерпана за часом, то буде виклик метода refreshToken, який перевірить поточний токен користувача та відправить у відповідь новий після його перевірки.

### 3.3 Розробка структури бази даних

Для зберігання даних в додатку використовується Система управління базами даних MySQL. В базі даних створюються таблиці згідно зі схемою, яка зображена на рисунку 3.5. Ця модель дозволяє описувати дані за допомогою загальновідомих блоків, які разом утворюють структуру.

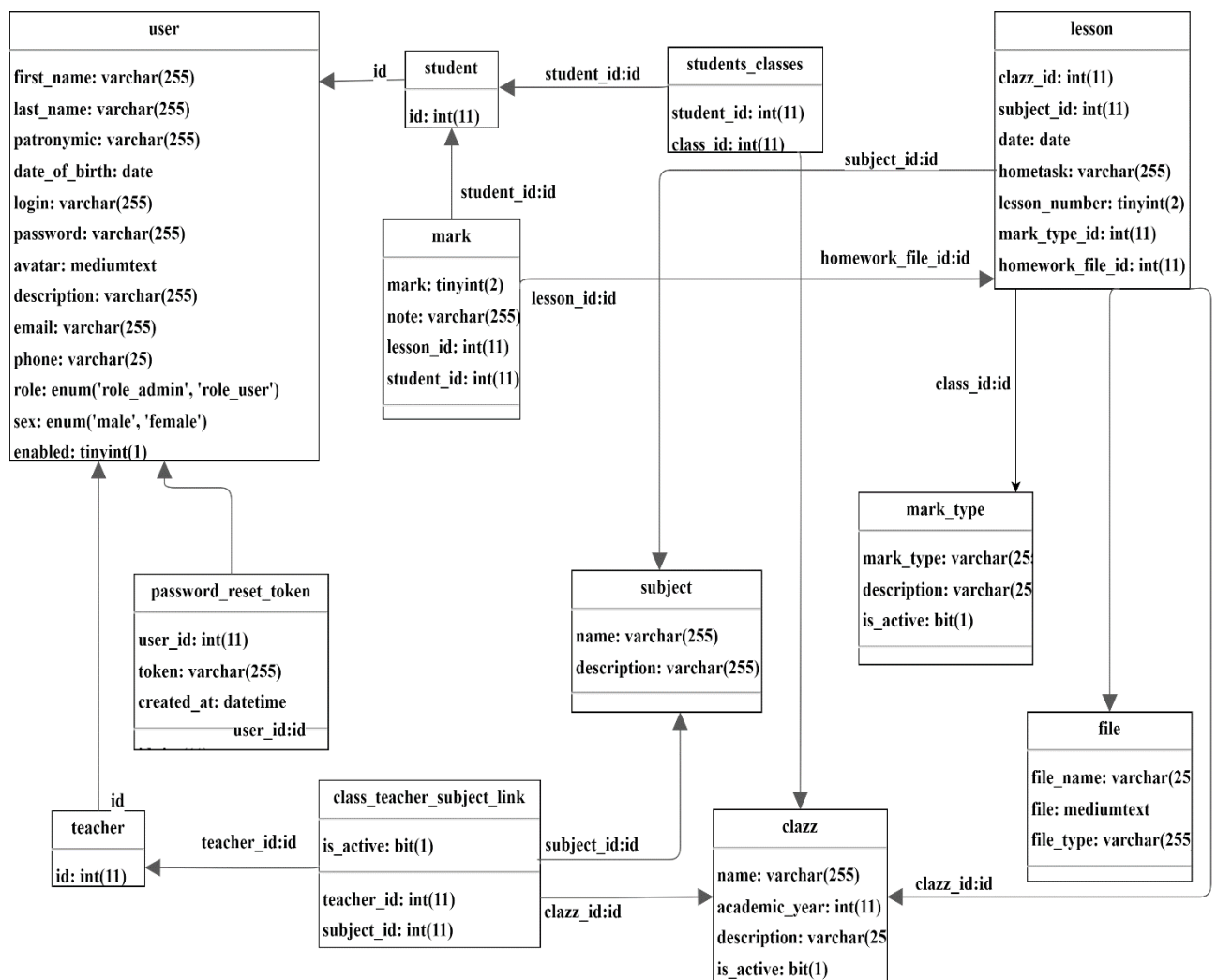


Рисунок 3.5 - ER-модель бази даних

Метою цієї діаграми є опис моделей даних, які можуть бути використані для опису предметної області. Вона надає зручний спосіб візуалізації зв'язків між різними компонентами цих моделей. Дана модель є універсальною, оскільки не залежить від конкретних засобів програмування, які будуть використовуватись для реалізації цих моделей.

У діаграмі використовуються різні типи зв'язків, такі як один до одного, один до багатьох і багато до багатьох, що допомагає краще розібратися в систематичному процесі опису предметної області. Крім того, дана модель включає ієрархічну, мережеву, об'єктну та реляційну моделі. Це означає, що будь-який з цих типів моделей може бути реалізований з використанням даної моделі.

Ієрархічна модель представлена у вигляді деревовидної структури, де об'єкти пов'язані між собою і мають різні типи зв'язків. Більшість об'єктів мають відношення "предок-нащадок", але можуть бути винятки, коли нащадки не мають предків або нащадок має багато предків.

Мережева модель є розширенням ієрархічної моделі, де нащадку може відповідати багато предків, відмінно від ієрархічної моделі, де нащадок має лише одного предка.

Модель, є реляційною, оскільки вона описана засобами, що базуються на її структурі і не містить додаткової інформації про фізичну організацію. Вона втілює три складові реляційної моделі даних: структурну, маніпуляційну та цілісну. Структурна частина вказує на те, що таблиці складаються з кортежів та атрибутів і містять відношення з іншими таблицями.

Маніпуляційна частина показує, що цими таблицями можна керувати за допомогою реляційної алгебри та числення. Цілісна частина гарантує, що всі кортежі унікальні, тобто таблиця має первинний ключ. Це дозволяє виконувати різноманітні обчислення, такі як додавання, віднімання, множення та ділення, над числовими даними, які містяться в таблицях.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

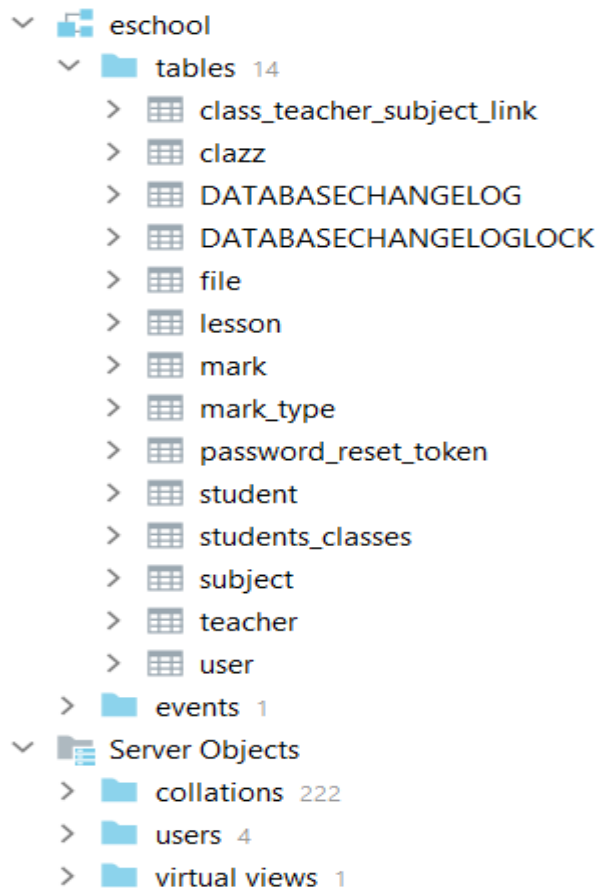


Рисунок 3.6 — База даних застосунку

На рисунку 3.6 зображена структура бази даних застосунку. На рисунку містяться таблиці та службова інформація.

### 3.4 Серверна частина

Мова програмування Java демонструє високу продуктивність та працює на платформах Windows і Linux. Вона має відкритий код і широко використовується для розробки сучасних хмарних застосунків. За допомогою цієї мови також можна створювати мобільні додатки та розгорнути їх в хмарі або локальному середовищі.

Для документування кінцевих точок використовується специфікація Rest API Swagger [14]. Архітектура Rest передбачає передачу стану в представленні і використовується як стиль взаємодії компонентів в мережі Інтернет.

Використання обмежень, накладених Rest, спрощує архітектуру додатку і покращує його ефективність.

На рисунку 3.7 зображено інтерфейс swagger, який описує кінцеві точки додатку та показує, що додаток містить автентифікацію та авторизацію (біля кожної кінцевої точки зображений замок).

admin-edit-user-controller : Admin Edit User Controller		Show/Hide	List Operations	Expand Operations
PUT	/admin/students/{idStudent}			updateStudent
PUT	/admin/teachers/{idTeacher}			updateTeacher
authentication-controller : Authentication Controller		Show/Hide	List Operations	Expand Operations
GET	/refresh			refreshAndGetAuthenticationToken
GET	/requestPasswordReset			recoverPassword
PUT	/resetPassword			updatePassword
POST	/signin			createAuthenticationToken

Рисунок 3.7 — Частина переліку кінцевих точок веб-додатку

На рисунку 3.8 показано вигляд роботи методу. При відкритті кінцевої точки "отримати класи" першим кроком потрібно ввести номер сторінки та кількість записів, які потрібно вибрати. Після натискання кнопки "Виконати" буде сформований запит curl [15], який включатиме різні заголовки, куки-файли, користувацький агент, посилання та інше. Вище буде відображена сформована URL-адреса цільового ресурсу разом із введеним запитом.

Кожна відповідь від сервера буде отримана у форматі текстового обміну даними JSON [14]. Під відповіддю JSON будуть міститися заголовки, статус коду відповіді, медіатип та приклад відповіді.

На креслинику IA94.190БАК.003 Д2 зображено діаграму діяльності з вхідними точками застосунку. Кожен процес перед початком роботи отримує дані від користувача, потім обробляє їх та надсилає відповідь з певним кодом, що означає результат відпрацювання запиту.

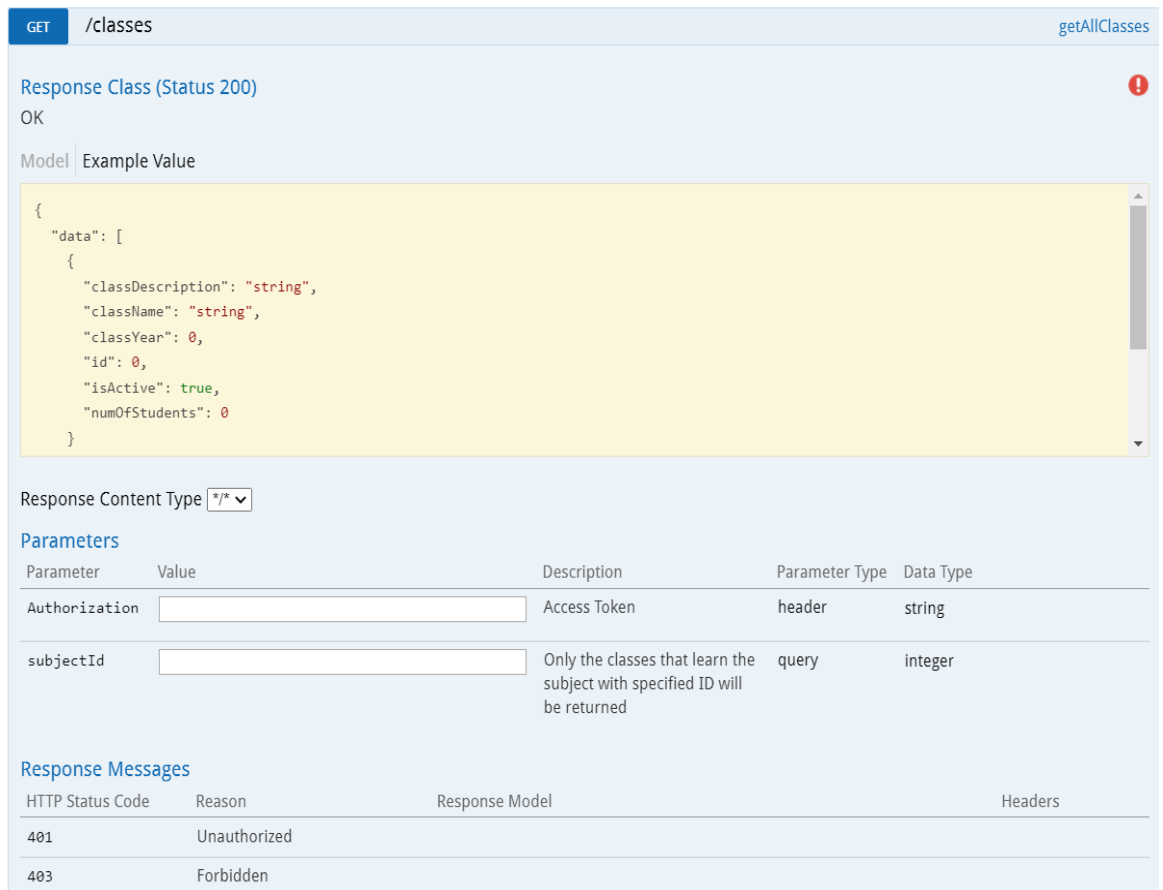


Рисунок 3.8 — Відпрацювання методу «отримати класи» у інтерфейсі swagger

Розглянемо програмну реалізацію кінцевої точки для отримання класів. На рисунку 3.9 показаний код методу `getAllClasses`. Метод контролера "ClassController" має атрибут "`@GetMapping`", який вказує, що цей метод використовує протокол HTTP та метод GET [15]. Далі, метод має параметр з атрибутом "`@RequestParam`", який вказує, що цей параметр буде взятий з URL-адреси.

Наступним кроком є метод сервісу "getAllClasses", який отримує параметр, виконує вибірку даних відповідно до вказаних параметрів та повертає екземпляр класу "Class" з типом "ClassDto". Клас "ClassServiceImpl" також містить всі необхідні методи та властивості, але додатково містить розширену інформацію про дані, їх кількість, загальну кількість записів та номер сторінки. Для цього створюється анонімний клас, який містить метадані про записи і додається до

заголовка відповіді. Також проводиться логування даних щодо кількості записів. Анонімний клас серіалізується та додається до заголовка.

Останнім кроком є мапінг даних, щоб відібрати необхідні дані для користувача, та формування типізованої відповіді зі статусом 200. Ця відповідь підтверджує успішне виконання операції.

```
@PreAuthorize("hasAnyRole('ADMIN', 'TEACHER')")
@GetMapping
public GeneralResponseWrapper<List<ClassDTO>> getAllClasses(
    @ApiParam(value = "Only the classes that learn the subject with specified ID will be returned")
    @RequestParam(required = false) Integer subjectId
){
    if (subjectId == null) {
        return new GeneralResponseWrapper<>(
            Status.of(HttpStatus.OK),
            classService.getAllClasses()
        );
    } else {
        return new GeneralResponseWrapper<>(
            Status.of(HttpStatus.OK),
            classService.getClassesBySubject(subjectId)
        );
    }
}
```

Рисунок 3.9 — Програма реалізація методу «getAllClasses»

На рисунку 3.10 можна побачити лог-повідомлення з докладною інформацією про початок запиту до бази даних. Це включає дані, такі як час початку запиту, інформацію про користувача, що здійснює запит, а також тип запиту, який виконується. Далі, у лозі відображається сформований запит SQL, який включає необхідні параметри для отримання даних. Цей запит може містити усі необхідні умови і фільтри, що вказуються для вибірки певних записів з бази даних.

Цей лог-вивід дозволяє слідкувати за всіма важливими деталями процесу отримання даних, включаючи початок запиту, параметри запиту та виконання SQL-запиту до бази даних. Він забезпечує додаткову інформацію та контроль над взаємодією з базою даних, що допомагає відлагоджувати та аналізувати роботу програми з точки зору доступу до даних.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

```

2023-05-25 17:49:37 [INFO ] liquibase :92 [localhost-startStop-1] - Successfully acquired change log lock
2023-05-25 17:49:38 [INFO ] liquibase :92 [localhost-startStop-1] - Creating database history table with r
2023-05-25 17:49:38 [INFO ] liquibase :92 [localhost-startStop-1] - Reading from eschool.DATABASECHANGELOG
2023-05-25 17:49:39 [INFO ] liquibase :92 [localhost-startStop-1] - classpath:db/liquibase-changelog.xml:
2023-05-25 17:49:39 [INFO ] liquibase :92 [localhost-startStop-1] - classpath:db/liquibase-changelog.xml:
2023-05-25 17:49:39 [INFO ] liquibase :92 [localhost-startStop-1] - classpath:db/liquibase-changelog.xml:
2023-05-25 17:49:39 [INFO ] liquibase :92 [localhost-startStop-1] - classpath:db/liquibase-changelog.xml:
2023-05-25 17:49:39 [INFO ] liquibase :92 [localhost-startStop-1] - classpath:db/liquibase-changelog.xml:
2023-05-25 17:49:39 [INFO ] liquibase :92 [localhost-startStop-1] - classpath:db/liquibase-changelog.xml:
2023-05-25 17:49:39 [INFO ] liquibase :92 [localhost-startStop-1] - classpath:db/liquibase-changelog.xml:
2023-05-25 17:49:39 [INFO ] liquibase :92 [localhost-startStop-1] - classpath:db/liquibase-changelog.xml:
2023-05-25 17:49:39 [INFO ] liquibase :92 [localhost-startStop-1] - classpath:db/liquibase-changelog.xml:
2023-05-25 17:49:39 [ERROR] liquibase :64 [localhost-startStop-1] - classpath:db/liquibase-changelog.xml:
ON SCHEDULE EVERY 5 MINUTE
STARTS CURRENT_TIMESTAMP
DO
DELETE FROM password_reset_token
WHERE ADDTIME(password_reset_token.created_at, "1:00:00") < now()
ON SCHEDULE EVERY 5 MINUTE
STARTS CURRENT_TIMESTAMP
2023-05-25 18:12:18 [INFO ] liquibase :92 [localhost-startStop-1] - Reading from eschool.DATABASECHANGELOG
2023-05-25 18:12:18 [INFO ] liquibase :92 [localhost-startStop-1] - classpath:db/liquibase-changelog.xml:
2023-05-25 18:12:18 [INFO ] liquibase :92 [localhost-startStop-1] - classpath:db/liquibase-changelog.xml:
2023-05-25 18:12:18 [INFO ] liquibase :92 [localhost-startStop-1] - classpath:db/liquibase-changelog.xml:
2023-05-25 18:12:18 [INFO ] liquibase :92 [localhost-startStop-1] - classpath:db/liquibase-changelog.xml:
2023-05-25 18:12:19 [INFO ] liquibase :92 [localhost-startStop-1] - classpath:db/liquibase-changelog.xml:
2023-05-25 18:12:19 [INFO ] liquibase :92 [localhost-startStop-1] - classpath:db/liquibase-changelog.xml:

```

Рисунок 3.10 — Логування додатку

### 3.5 Клієнтська частина

Клієнтська частина використовує сучасну мову програмування JavaScript.

Ця мова програмування використовується для створення динамічних інтерактивних веб-сторінок. Вона дозволяє веб-розробникам додавати різноманітні функції та ефекти до веб-сторінок, включаючи маніпуляцію елементами сторінки, обробку подій, валідацію форм, анімацію та багато іншого.

JavaScript є мовою, яка виконується безпосередньо в браузері, що дозволяє веб-сторінкам взаємодіяти з користувачем без необхідності завантаження нової сторінки з сервера. Вона є однією з найпопулярніших мов програмування, яка застосовується як на веб-сторонах, так і на серверах, що дозволяє створювати повноцінні веб-додатки та веб-сервіси.

JavaScript має широку підтримку серед браузерів і стандартизована шляхом специфікації ECMAScript, що забезпечує стабільну та сумісну роботу коду на різних платформах і браузерах.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

На рисунку 3.11 представлено, як виглядає розробка модулю реєстрації користувача на платформі.

```
<div class="row form-group required">
  <label for="fname" class="control-label col-4 requiredField font-weight-bold">Ім'я<span class="asteriskField">*</span></label>
  <div class="controls col-8">
    <input class="input-md textinput textInput form-control" id="fname" type="text" name="firstname">
  </div>
</div>
<div class="row form-group required">
  <label for="lname" class="control-label col-4 requiredField font-weight-bold">Прізвище<span class="asteriskField">*</span></label>
  <div class="controls col-8">
    <input class="input-md textinput textInput form-control" id="lname" type="text" name="lastname">
  </div>
</div>
<div class="row form-group required">
  <label for="patronymic" class="control-label col-4 requiredField font-weight-bold">По-батькові<span class="asteriskField">*</span></label>
  <div class="controls col-8">
    <input class="input-md textinput textInput form-control" id="patronymic" type="text" name="patronymic" required>
  </div>
</div>
```

Рисунок 3.11 – Розробка модулю «Особиста інформація»

На рисунку 3.12 демонструється, що баче користувач безпосередньо при переході на сторінку особистого профілю.

## Особиста інформація

Ім'я*	<input type="text" value="Катерина"/>
Прізвище*	<input type="text" value="Полянська"/>
По-батькові*	<input type="text" value="Валеріївна"/>
Логін*	<input type="text" value="kPolyan16"/>
Дата народження*	<input type="text" value="08/16/1999"/> <input type="button" value="📅"/>
Електронна адреса:	<input type="text"/>
Телефон:	<input type="text"/>
Старий пароль:	<input type="password"/>
Новий пароль:	<input type="password"/>
Повторіть новий пароль:	<input type="password"/>

Рисунок 3.12 – Веб-сторіна «Особиста інформація»

### Висновок до розділу 3

В даному розділу було описана архітектура ПЗ, для прочатку було описано структуру проекту, перелічений вміст папок та для яких функцій вони призначені. Також була наведена діаграма класів та описані зв'язки між класами.

Після чого була описана автентифікація, саме поннятя, було описано які види автентифікація існують. Також було описано який модуль відповідає за вхід в систему та перевірку логіна та пароля, що відпувається при вході та як генерується JWT токен, а також що відбувається коли закінчується сесія користувача.

Було описано створену базу даних, представлено діаграму таблиць БД та описано зв'язок між цими таблицями.

Була описана серверна частина, для документування кінцевих точок використовується специфікація Rest API Swagger. Архітектура Rest передбачає передачу стану в представленні і використовується як стиль взаємодії компонентів в мережі Інтернет. Було представлено переліку кінцевих точок веб-додатку та наведено відпрацювання методу «отримати класи» у інтерфейсі swagger, а також логування додатку.

На останок було продемонстровано роботу клієнтської частини на прикладі розробки модулю «Особиста інформація».

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

## 4 РОЗГОРТАННЯ ТА ТЕСТУВАННЯ ЗАСТОСУНКУ

### 4.1 Розгортання сервісу

Серверна частина застосунку розгортається, щоб система була готовою до використання, необхідно провести розгортання [14]. Розгортання є необхідною частиною життєвого циклу програмного забезпечення.

Docker - це платформа для доставки, виконання та керування додатками. Docker допомагає встановити всі залежності системи і упакувати їх у зображення, яке можна запустити в контейнерах. Це спрощує керування інфраструктурою застосунку. Використання даної технології дозволяє скоротити час розгортання застосунку.

Упакований застосунок запускається в ізольованому контейнері, відокремленому від середовища. Це підвищує безпеку додатку. Контейнери застосунку можуть виконуватися на одному хості, тобто на пристрої, підключеному до мережі, з унікальною адресою.

Файл конфігурації зображення застосунку повинен знаходитись у робочій директорії. В ньому описується послідовність дій, які Docker повинен виконати для розгортання застосунку. Тут можуть бути інструкції, які завантажують додаткові зображення з локальних або публічних директорій. Інструкції визначають робочу директорію у зображенні, побудованому на платформі Linux [14]. Контейнеру присвоюється файлова система для внесення змін. Створюється мережевий інтерфейс контейнера. Файли копіюються в зображення, виконуються додаткові команди для відновлення та конфігурації. У кінці виконується команда, що запускає застосунок.

Для керування контейнерами використовується інструмент Compose [14]. Цей інструмент дозволяє запускати контейнери однією командою. За допомогою системи контролю версій можна керувати кожним екземпляром застосунку у різних контейнерах.

Docker використовує готові зображення замість створення нових

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

кожного разу, що дозволяє економити пам'ять. Також контейнери можна пере використовувати. Зміни у застосунку швидко відображаються в запущеному контейнері.

Змінні середовища можуть бути використані в файлі конфігурації та самому застосунку, вказавши їх у файлі `Yaml` [15]. Ці змінні використовуються, наприклад, для рядка підключення до контейнера з базою даних або для конфігураційних даних системи та користувача.

Файл конфігурації застосунку також може містити деякі параметри, які можна налаштувати залежно від потреби. Замість жорсткого за кодування цих значень в самому коді, можна використовувати змінні середовища, що дозволяє змінювати їх без необхідності змінювати код. Наприклад, одна з таких змінних може використовуватись для зберігання певних значень для шифрування ідентифікаторів бази даних. З цим підходом, якщо конфігурація бази даних зміниться, достатньо буде змінити значення цієї змінної середовища, не змінюючи сам код застосунку.

Крім того, змінні середовища можуть бути використані для зберігання конфігураційних даних системи або користувача. Наприклад, можна мати змінну середовища, яка визначає шлях до папки з файлами користувача або параметри підключення до зовнішніх служб. Це дозволяє зручно налаштувати застосунок для кожного середовища окремо без потреби вручну змінювати значення в коді.

Використання змінних середовища в файлі конфігурації та самому застосунку дозволяє зробити його більш гнучким, налаштовуваним та переносимим. Змінні середовища використовуються для забезпечення розділення конфігурації від коду, полегшення налаштування та підтримки додатка в різних середовищах.

На рисунку 4.1 показані робочі контейнери додатку. Якщо не вказати явно, `Docker` може надати кожному контейнеру ім'я. Із цього зображення видно ідентифікатори контейнерів, використані зображення, команди для запуску контейнерів, час їх створення і порти, які були визначені.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

```
PS C:\eschool> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS      PORTS                               NAMES
0b88e85865ff  mysql:5.7 "docker-entrypoint.s..." 11 days ago    Up 33 minutes  0.0.0.0:3306->3306/tcp, 33060/tcp  eschool-master-db-1
```

Рисунок 4.1 – Інформація про контейнери додатку

## 4.2 Тестування роботи застосунку

Після того як адміністратор створив обліковий запис, користувач має змогу зайти на веб-сайт використовуючи форму входу, як зображено на рисунку 4.2.

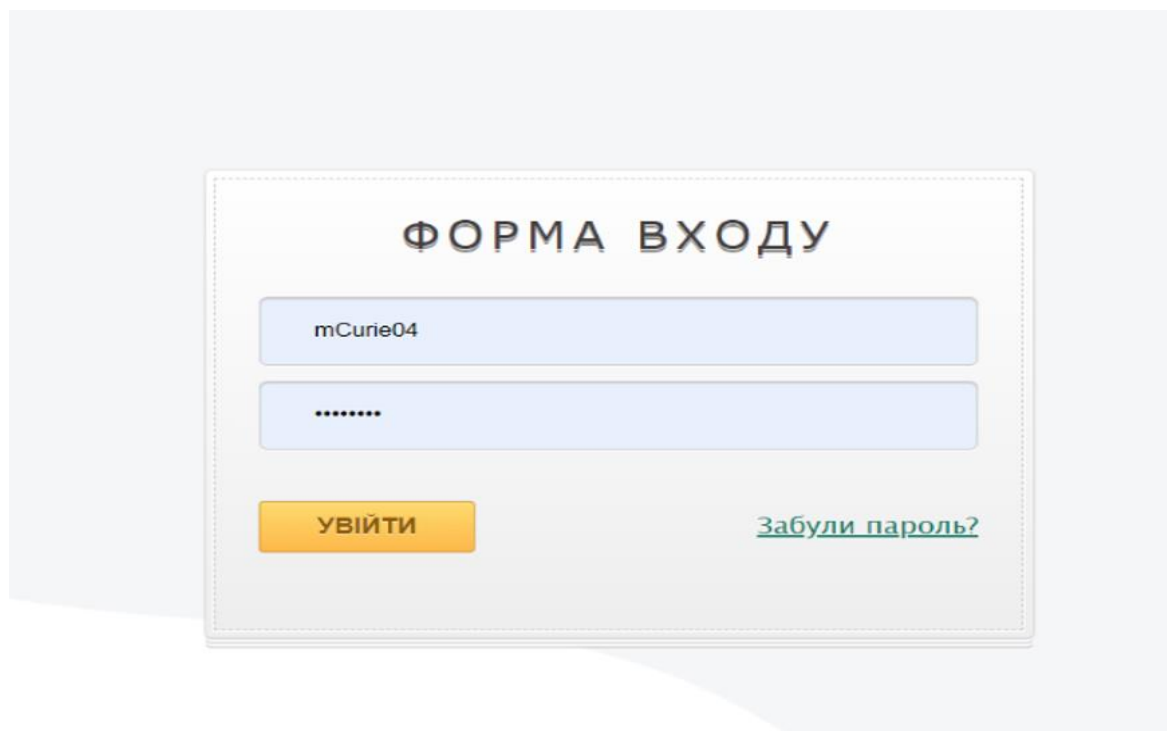


Рисунок 4.2 – Вхід в систему

Оскільки в застосунку наявний розділ ролей, користувач здійснює вхід в систему як адміністратор, вчитель або учень. На рисунку 4.3 зображено вхід в систему, як вчитель після коректного вводу логіна і пароля. На веб-сторінці відображені ініціали вчителя, який премент викладає. Також наявна можливість редагування профілю. На креслинику IA94.190БАК.003 Д4 зображено діаграму прецедентів з описом можливоїтей кожного користувача. Можна побочити, що дії користувачів можуть залежати від певних змін даних від інших користувачів.

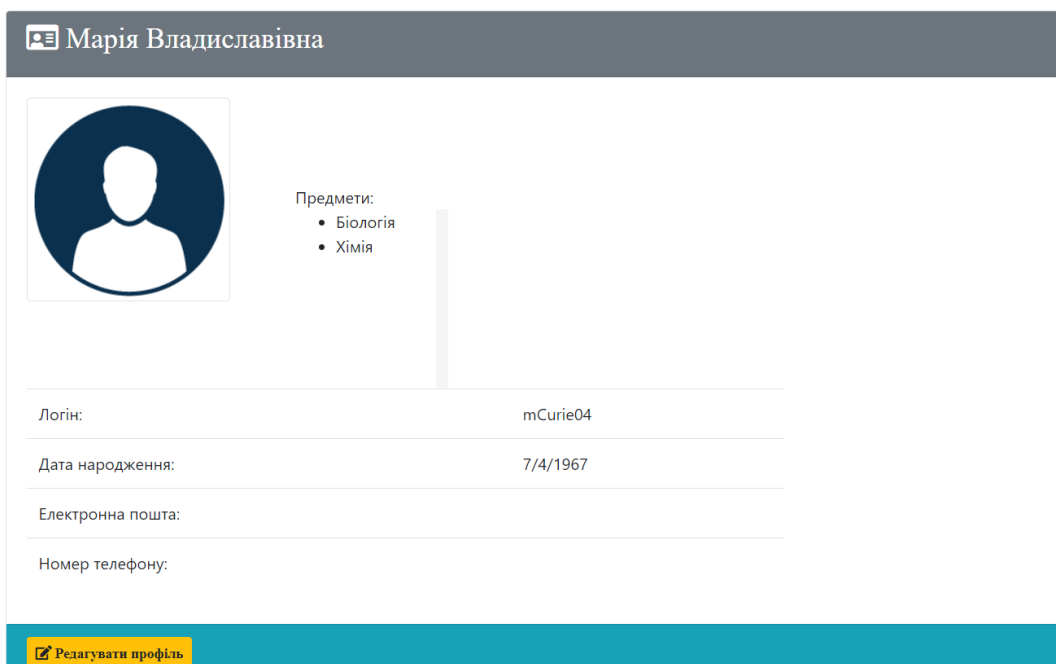


Рисунок 4.3 – Вхід в ситстему за роллю вчитель

На рисунку 4.4 демонструється початкова сторінка на якій можна додавати або переглянути список учнів та вчителів, додати та переглядати список предметів, створювати нові класи та переглядати існуючі, вести журнал, переглядати та виставляти в ньому оцінки, виводити статистику та переглядати розклад.

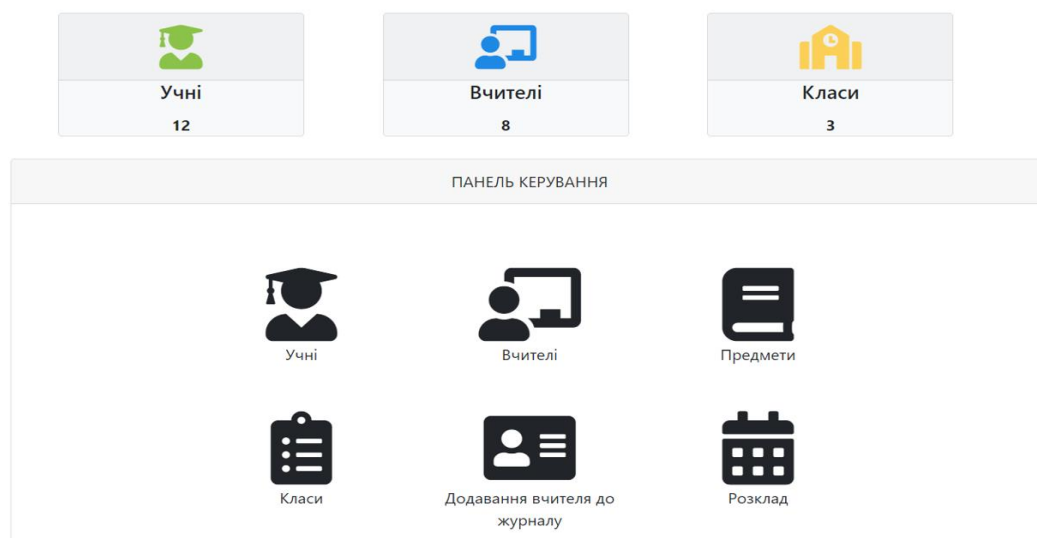


Рисунок 4.4 – Початкова сторінка

На рисунку 4.5 зображено створення нового учня, заповнення необхідних полей для створення облікового запису.

## Створення профілю учня

**Особиста інформація**

Прізвище*	<input type="text" value="Василь"/>
Ім'я*	<input type="text" value="Василь"/>
По-батькові*	<input type="text" value="Василь"/>
Логін	<input type="text" value="vasyl2023"/>
Клас*	<input type="text" value="9"/>
Дата народження*	<input type="text" value="02/02/2010"/>
Електронна адреса:	<input type="text" value="vasyl2023@mail.com"/>
Телефон:	<input type="text" value="+380888888888"/>

Рисунок 4.5 – Створення профілю учня

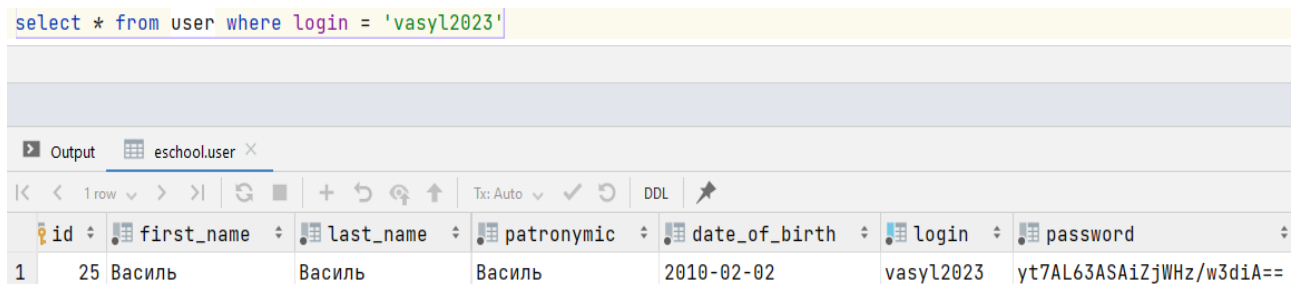
Після успішного створення профілю учня, він буде відображатись в відповідном класі, в нашуму випадку у 9 класі, що зображено на рисунку 4.6.

	<h3>Учні 9 класу (2018)</h3>	
	<a href="#">Маргарита Висока</a>	
	<a href="#">Ігор Підгорний</a>	
	<a href="#">Віта Квітовська</a>	
	<a href="#">Степан Когут</a>	
	<a href="#">Зеновій Запухляк</a>	
	<a href="#">Тетяна Куца</a>	
	<div style="border: 1px solid red; padding: 2px;"><a href="#">Василь Василь</a></div>	
	<input type="button" value="Надіслати дані учнів на пошту"/>	

Рисунок 4.6 – Прегляд учнів відповідного класу

Також після створення нового профілю всі дані заносяться в БД. На рисунку 4.7 зображено запит в БД по логіну учня.

```
select * from user where login = 'vasyl2023'
```



id	first_name	last_name	patronymic	date_of_birth	login	password	
1	25	Василь	Василь	Василь	2010-02-02	vasyl2023	yt7AL63ASAiZjWHz/w3diA==

Рисунок 4.7 – Запит в БД

На рисунку 4.8 представлено додавання нового класу. Для створення класу необхідно ввести назву класу та навчальний рік, а також є можливість надати додаткової інформації.

## Додати клас

Назва класу\*

11A

Навчальний рік\*

2023

Додаткова  
інформація

Клас з поглибленим вивченням іноземних мов.

Активний

Неактивний

Зберегти

Рисунок 4.8 – Додання нового класу

На рисунку 4.9 демонструється успішне додавання класу.

## Редагувати клас

7-A 2018  
7-B 2018  
9 2018  
11A 2023

Додати клас

Неактивні класи +

Рисунок 4.9 – Перегляд класів

Також є можливість редагувати предмети, що зображено на рисунку 4.10.

## РЕДАГУВАННЯ ПРЕДМЕТУ

Назва предмету\* Інформатика

Опис предмету

Додати

Рисунок 4.10 – Редагування предмету

На рисунку 4.11 демонструється можливість перегляду всіх предметів, пошуку необхідного предмету, а також можливість додавати новий предмет.

## ВСІ ПРЕДМЕТИ

Пошук

1	Історія України
2	Інформатика
3	Англійська мова
4	Українська мова
5	Українська література
6	Фізика
7	Географія
8	Біологія
9	Математика
10	Хімія

Додати предмет

Рисунок 4.11 – Перегляд всіх предметів

Також в застосунку наявний зручний журнал, в якому можна переглядати оцінки для обраного класу, за обраним предметом та навчальним роком. Заповнений журнал можна побачити на ристунку 4.12.

School Management System		Ви увійшли як Адміністратор Вийти													
Клас: 7-А				Предмет: Англійська мова						Навчальний рік: 2018/2019					
Тип оцінки	Кон. ▾	Пра. ▾	Лаб. ▾	Мод. ▾	Лаб. ▾	Лаб. ▾	Пра. ▾	Пра. ▾	Пра. ▾	Пра. ▾	Пра. ▾	Мод. ▾	Мод. ▾	Пра. ▾	Пра. ▾
Дата	09.04	09.06	09.11	09.13	09.18	09.20	09.25	09.27	10.02	10.04	10.09	10.11	10.16	10.18	10. С.
1.Ірина Грушецька		5	6		5						6	5	10		6.15
2.Катерина Полянська	4	7		11				9				5		8	7.67
3.Марія Василик			12			7	8	8				11	10		8.18

Студент:

Оцінка 1 ▾

Нотатка:

Рисунко 4.12 – Журнал учнів

На рисунку 4.13 продемонстровано можливість перегляду статистики за певний предмет, можна обрати клас в якому необхідно переглянути статистику та за певний навчальний рік.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

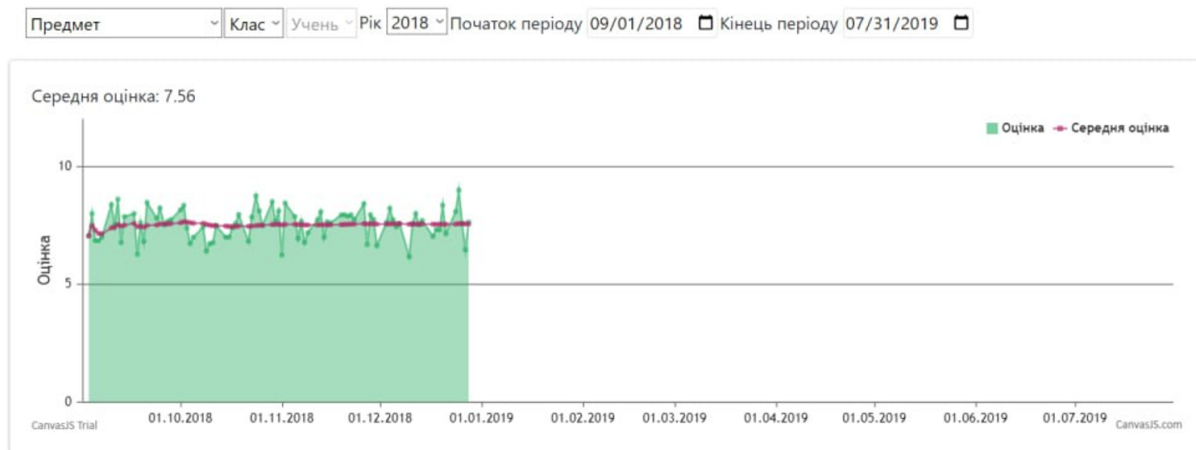


Рисунок 4.13 – Перегляд статистики

Графічне зображення статистики допомагає швидше зрозуміти зміни успішності учнів та побачити у який проміжок часу відбуваються позитивні та негативні тенденції, що в свою чергу допоможе розподілити навантаження у навчанні.

#### Висновок до розділу 4

В даному розділі було розглянуто як розгортається серверна частина застосунку, а саме з використанням технології Docker. Було надано інформацію про контейнери застосунку.

Після чого було протестовано роботу застосунку. Проілюстровано вхід в систему з роллю вчитель, зображено початкову сторінку, можливість додавання нового учня та продемонстровано збереження цього учня в БД. Також було додано новий клас та перевірено, що цей клас було успішно створено. Потім було продемонстровано можливість перегляду всіх предметів та можливість редагування. Після чого було наведено приклад журналу з оцінками та можливість перегляду статистики.

В цілому, в даному розділі було описано функціонал, який має створений застосунок.

## ВИСНОВКИ

В бакалаврському проєкті було проаналізовано існуючі рішення та створено адаптивну платформу для персоналізованого онлайн-навчання.

У першому розділі проводився аналіз існуючих рішень на ринку онлайн-навчання. В результаті аналізу було виявлено деякі недоліки існуючих додатків, такі як обмежений функціонал, незручний інтерфейс користувача, складність використання тощо. Зроблено висновок, що розробка нового застосунку є доцільною, оскільки він дозволить вирішити ці недоліки та забезпечити більш зручне та ефективне навчання.

У другому розділі було ознайомлено з мовою програмування та технологіями, які були обрані для розробки даного застосунку. Java – обрана мова програмування, Spring Boot, Spring MVC, Spring Security, Hibernate – використані фреймворки, MySQL – база даних для збереження інформації, Swagger використаний для документації, а також HTML і JavaScript для клієнтської частини.

У третьому розділі було описано розробку програмного забезпечення, продемонстрована структура проєкту та ієрархія класів. Продемонстровано принцип роботи, автентифікація, зображено структу БД та вміст таблиць. А також ознайомлено з клієнтською та серверною частинами.

Четвертий розділ присвячений розгортанню проєкту та детальному опису його функціоналу. Показано процес розгортання платформи на сервері і встановлення необхідних залежностей. Детально описані функціональні можливості платформи, такі як створення класів, управління користувачами, доступ до статистики тощо.

В результаті виконання бакалаврського проєкту були досягнуті всі поставлені цілі. Розроблений зручний та функціональний застосунок, який має потенціал для використання в навчальному процесі шкіл та університетів. Його адаптивна реалізація дозволяє користувачам налаштовувати навчання під свої потреби та отримувати індивідуальну підтримку. Застосунок може стати цінним інструментом для вдосконалення освітнього процесу.

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Knewton Adaptive Learning Technology [Електронний ресурс] – Режим доступу до ресурсу:  
<https://voices.uchicago.edu/201702busn3910001/2017/04/07/knewton-adaptive-learning-technology/>
2. How Google Conquered the Classroom: The Googlification of Schools Worldwide [Електронний ресурс] – Режим доступу до ресурсу:  
<https://research.com/education/how-google-conquered-the-classroom>
3. An Overview of Google Classroom [Електронний ресурс] – Режим доступу до ресурсу: <https://innovation-village.com/an-overview-of-google-classroom/>
4. History of Java [Електронний ресурс] – Режим доступу до ресурсу:  
<https://www.javatpoint.com/history-of-java>
5. The complete History of Java Programming Language [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/the-complete-history-of-java-programming-language/>
6. Spring Boot – Introduction [Електронний ресурс] – Режим доступу до ресурсу:  
[https://www.tutorialspoint.com/spring\\_boot/spring\\_boot\\_introduction.htm](https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm)
7. Spring - MVC Framework [Електронний ресурс] – Режим доступу до ресурсу:  
[https://www.tutorialspoint.com/spring/spring\\_web\\_mvc\\_framework.htm](https://www.tutorialspoint.com/spring/spring_web_mvc_framework.htm)
8. Introduction To Spring MVC Flow Diagram & Advantages [Електронний ресурс] – Режим доступу до ресурсу: <https://www.upgrad.com/blog/spring-mvc-flow-diagram/>

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

- 9 What Is Spring Security? [Електронний ресурс] – Режим доступу до ресурсу:  
<https://www.developer.com/design/what-is-spring-security/>
10. What is Spring security [Електронний ресурс] – Режим доступу до ресурсу: <https://www.javadevjournal.com/spring/what-is-spring-security/>
11. Spring Security Overview [Електронний ресурс] – Режим доступу до ресурсу: <https://auth0.com/blog/spring-security-overview/>
12. Hibernate [Електронний ресурс] – Режим доступу до ресурсу:  
<https://www.theserverside.com/definition/Hibernate#:~:text=Hibernate%20is%20an%20open%20source%20Object%2DRelational%20Persistence%20and%20Query,da%20persistence%20related%20programing%20tasks>
13. What is MySQL? [Електронний ресурс] – Режим доступу до ресурсу:  
<https://www.oracle.com/mysql/what-is-mysql/>
14. Сторінка документації «Docker» [Електронний ресурс] – Режим доступу:  
<https://docs.docker.com/>
15. Сторінка документації «Compose file» [Електронний ресурс] – Режим доступу:  
<https://docs.docker.com/compose/compose-file/compose-file-v3/>

					ІА94.190БАК.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

## ДОДАТОК А

Адаптивна платформа для персоналізованого онлайн-навчання

Програмний код

<https://github.com/cdDenys/DiplomaProject>

