

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

МУЛЬТИМЕДІЙНІ ІНТЕРФЕЙСИ ТА 3D ВІЗУАЛІЗАЦІЯ: РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ КУРСОВОГО ПРОЄКТУ

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для студентів,
які навчаються за спеціальністю F2 Інженерія програмного забезпечення
за освітньою програмою «Програмне забезпечення мультимедійних та
інформаційно-пошукових систем»*

Київ
КПІ ім. Ігоря Сікорського
2026

Мультимедійні інтерфейси та 3D візуалізація: рекомендації до виконання курсового проєкту [Електронний ресурс] : навч. посіб. для студентів спеціальності F2 Інженерія програмного забезпечення, освітня програма «Програмне забезпечення мультимедійних та інформаційно-пошукових систем» / КПІ ім. Ігоря Сікорського ; уклад.: Є.С. Сулема, О.К. Сулема. – Електронні текстові дані (1 файл: 2,01 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2026. – 64 с.

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 4 від 05.02.2026 р.)

за поданням Вченої ради факультету програмних систем та прикладної математики (протокол № 8 від 26.01.2026 р.)

Електронне мережне навчальне видання

МУЛЬТИМЕДІЙНІ ІНТЕРФЕЙСИ ТА 3D ВІЗУАЛІЗАЦІЯ:

РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ КУРСОВОГО ПРОЄКТУ

Укладачі: Сулема Євгенія Станіславівна, д-р техн. наук, доц.
Сулема Ольга Костянтинівна, д-р філ.

Відповідальний редактор Заболотня Тетяна Миколаївна, канд. техн. наук, доц.

Рецензент Гаврилко Євген Володимирович, д-р техн. наук, проф.

Навчальний посібник розроблено для ознайомлення студентів з вимогами та рекомендаціями до виконання курсового проєкту з дисципліни «Мультимедійні інтерфейси та 3D візуалізації». Навчальне видання призначене для студентів, які навчаються у магістратурі за спеціальністю F2 Інженерія програмного забезпечення, за освітньою програмою «Програмне забезпечення мультимедійних та інформаційно-пошукових систем» факультету програмних систем та прикладної математики КПІ ім. Ігоря Сікорського.

© КПІ ім. Ігоря Сікорського, 2026

ЗМІСТ

ВСТУП.....	5
1. ОСНОВНІ ВІДОМОСТІ ТА ВИМОГИ.....	6
1.1. Типове завдання	6
1.2. Приклади тематики курсових проєктів	6
1.3. Вимоги до мультимедійного помічника.....	7
2. ОРГАНІЗАЦІЯ ВИКОНАННЯ ТА ЗАХИСТУ КУРСОВОГО ПРОЄКТУ	11
2.1. Формування команди та розподіл ролей	11
2.2. Інструментарій командної роботи.....	13
2.3. Графік виконання робіт.....	13
2.4. Процедура захисту	14
3. ВИМОГИ ДО ДОКУМЕНТАЦІЇ	15
3.1. Технічне завдання	15
3.2. Пояснювальна записка	16
3.3. Керівництво користувача	16
3.4. Презентація та демонстраційні матеріали.....	17
4. ТЕОРЕТИЧНІ ОСНОВИ ПРОЄКТУВАННЯ МУЛЬТИМЕДІЙНИХ ІНТЕРФЕЙСІВ.....	18
4.1. Психологія сприйняття мультимедійної інформації.....	18
4.2. Особливості проєктування антропоморфних інтерфейсів	19
4.3. Принципи побудови людино-машинного діалогу.....	20
5. МЕТОДИКА ПРОЄКТУВАННЯ ГОЛОСОВОЇ ВЗАЄМОДІЇ	23
5.1. Сценарне проєктування (Scripting)	23
5.2. Робота з намірами (Intents) та сутностями (Slots)	24
5.3. Візуалізація логіки діалогу	25

5.4. Формування навчальної вибірки	25
5.5. Мультимодальна синхронізація	26
6. ТЕХНОЛОГІЇ 3D-ВІЗУАЛІЗАЦІЇ ТА АНІМАЦІЇ У ВЕБ-СЕРЕДОВИЩІ.....	27
6.1. Створення та оптимізація 3D-моделей для Web	27
6.2. Анімація персонажа.....	28
6.3. Автоматичний ліпсінк (Lip Sync).....	29
6.4. Програмна інтеграція (Three.js).....	30
7. ПРОГРАМНА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ	31
7.1. Архітектура системи.....	31
7.2. Реалізація клієнтської частини (Frontend).....	31
7.3. Реалізація серверної частини (Backend)	32
7.4. Інтеграція з хмарними сервісами (Cloud Integration)	33
7.5. Тестування системи	34
ДОДАТКИ.....	35
Додаток А. Приклад титульного аркуша.....	36
Додаток Б. Приклад технічного завдання	37
Додаток В. Приклад пояснювальної записки	41
Додаток 4. Приклад керівництва користувача.....	52
Додаток 5. Приклад презентації	59
ПОДЯКА	64

ВСТУП

Стрімкий розвиток інформаційних технологій докорінно змінює способи взаємодії людини з комп'ютерними системами. Епоха командного рядка та традиційних графічних інтерфейсів (GUI), що вимагали від користувача тривалого навчання й адаптації, поступово поступається місцем природним інтерфейсам користувача (Natural User Interfaces – NUI). Сучасні програмні продукти орієнтуються на інтуїтивну взаємодію, прагнучи «розуміти» людину через голос, жести, міміку та інші мультимодальні засоби комунікації.

У цих умовах особливої актуальності набуває підготовка фахівців, здатних розробляти програмне забезпечення з використанням мультимедійних інтерфейсів і засобів 3D-візуалізації. Виконання курсового проєкту з дисципліни «Мультимедійні інтерфейси та 3D-візуалізація» спрямоване на формування у студентів компетентностей, необхідних для розв'язання практичних завдань професійної діяльності в цій сфері.

Метою курсового проєкту є розвиток здатності самостійно проєктувати та реалізовувати програмні системи, що поєднують мультимедійні інтерфейси користувача та елементи 3D-візуалізації. Предметом проєктування виступають апаратні й програмні засоби подання, оброблення, перетворення та відтворення мультимедійної інформації.

У навчальному посібнику наведено опис типового завдання курсового проєктування, визначено вимоги та рекомендації щодо його виконання і захисту, а також вимоги до супровідної документації. Окрему увагу приділено теоретичним засадам проєктування мультимедійних інтерфейсів, методиці організації голосової взаємодії, технологіям 3D-візуалізації й анімації у веб-середовищі та програмній реалізації основних компонентів системи.

1. ОСНОВНІ ВІДОМОСТІ ТА ВИМОГИ

1.1. Типове завдання

Типове завдання для курсового проєктування з дисципліни «Мультимедійні інтерфейси та 3D-візуалізація» полягає у розробленні голосового інтерфейсу та мультимедійного помічника для обраної програмної системи.

Ключовою ідеєю є надання користувачеві альтернативного і природнішого способу взаємодії з вебзастосунком. Замість того, щоб лише натискати кнопки та вводити текст у поля пошуку, користувач може спілкуватися із системою голосом, отримуючи візуальний та аудіальний зворотний зв'язок від анімованого 3D-персонажа.

У такій системі пошукові запити або команди надаються користувачем у формі голосових повідомлень, які являють собою завершені речення українською або англійською мовою. Система повинна розпізнати мовлення, виділити з нього ключові слова (інтенцію та параметри) та виконати відповідну дію.

1.2. Приклади тематики курсових проєктів

Розглянемо декілька прикладів тематики, в рамках якої можна сформулювати індивідуальне завдання.

Розумний будинок (Smart Home Dashboard)

Візуалізація: 3D-план кімнати або будинку з інтерактивними об'єктами (лампи, термостати, жалюзі).

Голосові команди: «Увімкни світло на кухні», «Зроби температуру 22 градуси», «Відкрий штори».

Роль асистента: віртуальний консьєрж, що підтверджує виконання команд.

Конфігуратор товарів (E-commerce Configurator)

Візуалізація: детальна 3D-модель товару (автомобіль, кросівки, меблі), яку можна обертати.

Голосові команди: «Покажи у червоному кольорі», «Зміни диски на спортивні», «Скільки це коштує?».

Роль асистента: продавець-консультант, який описує переваги обраної комплектації.

Туристичний гід (Virtual Tour Guide)

Візуалізація: стилізована 3D-мапа міста або детальна модель історичної пам'ятки.

Голосові команди: «Проклади маршрут до замку», «Розкажи історію цієї будівлі», «Де найближча кав'ярня?».

Роль асистента: історичний персонаж або гід, що проводить екскурсію.

Навчальна система (EdTech Assistant)

Візуалізація: інтерактивні навчальні матеріали (наприклад, 3D-модель Сонячної системи або анатомії людини).

Голосові команди: «Покажи Марс», «З чого складається серце?», «Почни тест».

Роль асистента: вчитель або ментор, що пояснює матеріал та перевіряє знання.

1.3. Вимоги до мультимедійного помічника

Для підвищення комунікативного рівня програмної системи графічний інтерфейс має включати анімоване зображення мультимедійного помічника – персонажа, який «спілкується» з користувачем. Проектування цього компонента вимагає комплексного підходу, що охоплює як візуальну частину, так і логіку діалогу.

Вимоги до персонажа

Типаж

Персонажем може бути людина, тварина, робот або будь-яка інша віртуальна істота. Вибір персонажа має відповідати темі проєкту (наприклад, сова для бібліотеки, робот для ІТ-системи).

Синхронізація мовлення (Lip-Sync)

Зображення мультимедійного помічника має бути синхронізоване з його мовленням. Рухи губ (віземи) повинні відповідати звукам, що вимовляє асистент.

Стани анімації

Асистент не повинен бути статичним. Необхідно реалізувати щонайменше 4 стани:

- Idle (Спокій): персонаж «дихає», кліпає очима, чекає на команду.
- Listening (Слухання): візуальна реакція на те, що мікрофон увімкнено (нахил голови, прикладання руки до вуха тощо) (рис. 2).
- Thinking/Processing (Обробка): стан очікування відповіді від серверу.
- Speaking (Мовлення): активна жестикуляція та артикуляція.

Приклад графічного інтерфейсу зі зображенням мультимедійного помічника наведено на рис. 1.1.

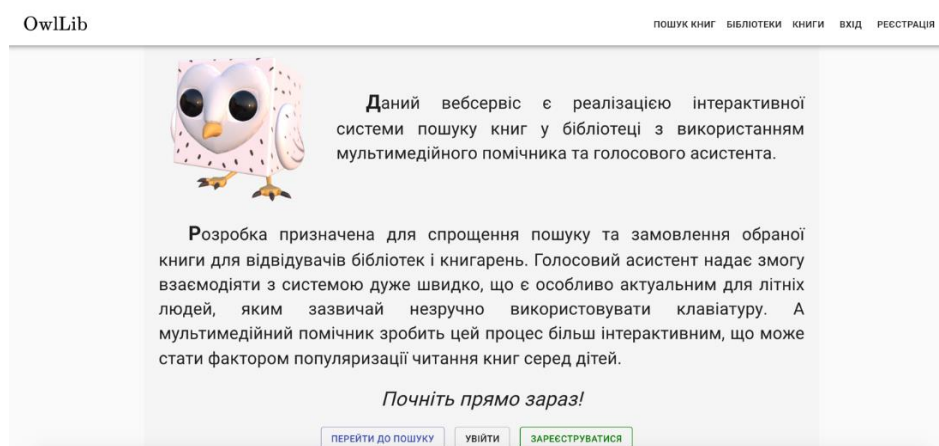


Рис. 1.1. Приклад інтерфейсу з мультимедійним помічником

Приклад синхронізації моделі мультимедійного помічника з процесом отримання голосових команд наведено на рис. 1.2.



Рис. 1.2. Анімація прослуховування

Мультимедійний інтерфейс має передбачати розгорнуті діалоги між користувачем та програмною системою. Діалоги повинні охоплювати велику кількість практичних ситуацій використання голосового інтерфейсу.

Для формалізації діалогів має бути розроблений сценарій взаємодії користувача та програмної системи. Цей процес складається з наступних кроків

Аналіз потреб та контексту

Спочатку потрібно визначити потреби користувачів системи та цільовий пристрій (планшет, смартфон, персональний комп'ютер тощо), на якому буде використовуватись програмний продукт. Від форм-фактора пристрою залежить довжина реплік та розташування візуальних елементів.

Розроблення сценарію

Після визначення меж голосової взаємодії створюється сценарій, який описує всі можливі гілки розмови.

Написання діалогів

На основі сценарію необхідно написати детальні діалоги (скрипти), включаючи варіанти фраз користувача та відповіді системи.

Робота над візуалізацією має здійснюватися паралельно з технічною реалізацією. Одночасно зі створенням анімацій виконується проєктування архітектури системи та її розроблення (backend/frontend). Завершується робота інтеграцією всіх компонентів системи: підключенням 3D-моделі до веб-інтерфейсу та налаштуванням її взаємодії з модулями розпізнавання та синтезу мовлення.

2. ОРГАНІЗАЦІЯ ВИКОНАННЯ ТА ЗАХИСТУ КУРСОВОГО ПРОЄКТУ

Виконання курсового проєкту – це не лише перевірка технічних навичок, а й симуляція реального робочого процесу в ІТ-компанії. Студенти вчаться працювати в команді, розподіляти відповідальність, дотримуватися дедлайнів та використовувати системи контролю версій.

2.1. Формування команди та розподіл ролей

Курсовий проєкт виконується групою студентів (звичайно 4 особи). Успіх проєкту залежить від чіткого розмежування зон відповідальності. Нижче наведено рекомендований розподіл ролей із деталізацією завдань (Job Descriptions), що розширює типовий розподіл:

Роль 1: Team Lead / Backend Developer (Архітектура та інтеграція)

Відповідає за: загальну архітектуру системи, серверну частину, базу даних та збірку проєкту.

Завдання:

- налаштування репозиторію (GitHub/GitLab) та правил комітів (Git Flow);
- проєктування REST API або GraphQL схеми;
- розроблення серверної логіки (Node.js/Python) для роботи із сутностями (книги, товари);
- інтеграція всіх компонентів системи у фінальний білд;
- комунікація з викладачем та контроль графіку виконання.

Роль 2: Frontend / VUI Developer (Голосовий інтерфейс)

Відповідає за: клієнтську частину веб-додатку, логіку розпізнавання голосу та UI.

Завдання:

- верстка графічного інтерфейсу (React/Vue/HTML+CSS);

- реалізація логіки Web Speech API або інтеграція хмарних сервісів (STT/TTS);
- програмування станів мікрофона (слухає, обробляє, говорить);
- візуалізація діалогів (субтитри, підказки).

Роль 3: 3D Artist / Animator (Графічне моделювання)

Відповідає за: візуальний образ асистента, анімацію та 3D-сцену.

Завдання:

- створення або оптимізація 3D-моделі персонажа (Blender/Maya);
- ріггінг (створення скелета) та скінінг моделі;
- створення анімаційних кліпів (Idle, Wave, Thinking);
- створення Shape Keys (Blend Shapes) для ліпсінку (синхронізації губ);
- експорт моделі у формат glTF/GLB.

Роль 4: Content Manager / QA Engineer (Сценарій та тестування)

Відповідає за: наповнення системи даними, логіку діалогу та якість роботи.

Завдання:

- розроблення сценарію взаємодії та деревоподібної структури діалогів;
- підготовка навчальних фраз (Intents) для розпізнавання;
- наповнення бази даних контентом (описи книг, картинки);
- тестування сценаріїв (Manual Testing) та пошук помилок у розпізнаванні;
- підготовка документації користувача.

Якщо група менша за 4 особи, ролі об'єднуються (наприклад, Team Lead бере на себе і Backend, і частину Frontend).

2.2. Інструментарій командної роботи

Для ефективної взаємодії рекомендується використовувати наступні інструменти.

Системи контролю версій: обов'язкове використання Git. Кожен студент повинен мати історію комітів, що підтверджує його внесок.

Task-трекери: Trello, Jira або GitHub Projects для відстеження статусу завдань (To Do, In Progress, Done).

Дизайн-інструменти: Figma для проектування інтерфейсів перед початком верстки.

2.3. Графік виконання робіт

Дотримання графіку є критичним для допуску до захисту. Орієнтовні дедлайни наведено у табл. 2.1.

Таблиця 2.1

Етап	Термін виконання	Результат
1. Ініціалізація	до 15 вересня	Сформовано групи, обрано тему, розподілено ролі.
2. Проектування	до 20 вересня	Затверджено Технічне завдання (ТЗ). Розроблено макети UI та сценарій діалогу.
3. Прототипування	жовтень-листопад	Готова серверна частина, "сіра" 3D-модель (blockout), базове розпізнавання голосу.

Продовження табл. 2.1

Етап	Термін виконання	Результат
4. Бета-версія	до 5 грудня	Робочий застосунок з повною функціональністю, але можливими дрібними багами.
5. Фіналізація	до 20 грудня	Виправлення помилок, написання документації, підготовка презентації. Захист проєкту.

2.4. Процедура захисту

Захист курсового проєкту відбувається публічно перед комісією.

Складові оцінки

- *Якість програмного продукту*: стабільність роботи, відповідність ТЗ, якість 3D-моделі та адекватність реакцій асистента.
- *Демонстрація (Live Demo)*: група має продемонструвати роботу системи в реальному часі. Обов'язково показати:
 - голосовий пошук та отримання результату;
 - анімацію персонажа (синхронізацію з мовленням);
 - оброблення помилок (що буде, якщо сказати нісенітницю).
- *Якість документації*: повнота пояснювальної записки та відповідність стандартам.
- *Відповіді на питання*: кожен студент відповідає за свою частину. Наприклад, дизайнер повинен пояснити вибір полігональної сітки, а розробник – вибір бази даних.

3. ВИМОГИ ДО ДОКУМЕНТАЦІЇ

Написання технічної документації є невід’ємною частиною інженерної роботи. Курсовий проєкт складається з програмної системи та супровідних документів. Кожен виконавець готує власну частину текстової документації відповідно до своєї ролі.

Документація включає наступні складові:

- 1) титульний аркуш;
- 2) технічне завдання;
- 3) пояснювальна записка;
- 4) керівництво користувача.

3.1. Технічне завдання

Технічне завдання (ТЗ) визначає індивідуальні вимоги до системи. Це юридичний документ між «замовником» (викладачем) та «виконавцем» (студентом).

Структура ТЗ

- Найменування та галузь застосування: де і ким буде використовуватись продукт.
- Призначення розробки: яку проблему вирішує система (наприклад, "спрощення пошуку книг для людей з вадами зору").
- Вимоги до програмного продукту:
 - функціональні вимоги: список функцій (пошук, бронювання, реєстрація); рекомендується формулювати їх у форматі User Stories ("Як користувач, я хочу...");
 - вимоги до інтерфейсу: наявність 3D-персонажа, підтримка української мови.
- Етапи проєктування: календарний план з датами.

3.2. Пояснювальна записка

Пояснювальна записка (ПЗ) є основним документом, у якому студент описує, як саме було виконано завдання. ПЗ має демонструвати глибину технічних знань.

Типова структура пояснювальної записки

- Вступ: актуальність, мета, задачі.
- Аналіз інструментів розроблення: обґрунтування вибору технологій (чому React, а не Angular? Чому Three.js?).
- Розроблення архітектури і логіки:
 - діаграми класів, діаграми послідовності (Sequence Diagram) для голосових команд;
 - опис API та структури бази даних.
- Проектування мультимедійного інтерфейсу:
 - сценарій діалогу (скрипт);
 - опис станів анімації персонажа (Idle, Talk, Listen);
 - методи синхронізації губ (Lip-sync algorithms).
- Інтеграція компонентів: як Frontend з'єднується з Backend та Google Cloud API.
- Тестування: опис тест-кейсів (що перевіряли) та результатів (скріншоти).
- Висновки: чого досягли, які були проблеми, плани на майбутнє.
- Список джерел.

Текст пояснювальної записки має бути насичений діаграмами, фрагментами коду (з коментарями) та скріншотами.

3.3. Керівництво користувача

Цей документ призначений для кінцевого користувача, який не є програмістом. Він має наочно ілюструвати логіку роботи з системою.

Вимоги до керівництва користувача

- Візуалізація: велика кількість скріншотів інтерфейсу (рис. 3.1-3.3).
- Покрокові інструкції: алгоритм дій для вирішення конкретних задач.
 - Приклад: "Для замовлення книги натисніть кнопку мікрофона (рис. 1) та скажіть назву книги. Після появи результатів скажіть 'Замовити першу'".
- Опис помилок: що робити, якщо система не зрозуміла команду (наприклад, "Перевірте підключення до інтернету" або "Вимовляйте слова чіткіше").

3.4. Презентація та демонстраційні матеріали

Захист супроводжується презентацією (PowerPoint/Google Slides).

Структура слайдів

- Титульний слайд: назва теми, автори.
- Команда: фото та ролі учасників.
- Проблема та рішення: чому цей проєкт важливий.
- Технічне завдання: ключові функції.
- Технологічний стек: логотипи використаних інструментів (JS, Python, Blender тощо).
- Архітектура: схема взаємодії модулів.
- Деталі реалізації: найскладніші моменти (сітка 3D-моделі, алгоритм обробки звуку).
- Демонстрація: відео або скріншоти роботи.
- Висновки.

Приклади слайдів наведені у Додатку 5.

Презентація має бути лаконічною, текст на слайдах – тезовим. Під час виступу у жодному разі не можна читати текст зі слайдів. Слайди мають доповнювати виступ, а не дублювати його текст. Значна увага під час доповіді має бути приділена графікам та схемам, наведеним на слайдах.

4. ТЕОРЕТИЧНІ ОСНОВИ ПРОЄКТУВАННЯ МУЛЬТИМЕДІЙНИХ ІНТЕРФЕЙСІВ

Створення ефективного мультимедійного асистента вимагає не лише навичок програмування та 3D-моделювання, але й глибокого розуміння принципів взаємодії людини з комп'ютером (HCI – Human-Computer Interaction). У цьому розділі розглядаються психологічні аспекти сприйняття інформації, особливості створення антропоморфних інтерфейсів та принципи побудови діалогових систем.

4.1. Психологія сприйняття мультимедійної інформації

Мультимедійні інтерфейси базуються на використанні декількох каналів передачі інформації (модальностей) – візуального та аудіального. Розуміння того, як людина обробляє ці потоки, є критично важливим для створення зручної системи.

4.1.1. Теорія когнітивного навантаження

Когнітивне навантаження (Cognitive Load) – це обсяг зусиль робочої пам'яті, необхідний для виконання завдання. Робоча пам'ять людини має обмежену ємність. При проєктуванні голосових інтерфейсів з 3D-аватаром розробник повинен враховувати три типи навантаження.

1. **Внутрішнє (Intrinsic):** складність самої задачі (наприклад, вибір книги за багатьма критеріями). Цей рівень залежить від предметної області.
2. **Зовнішнє (Extraneous):** навантаження, створене невдалим дизайном інтерфейсу (наприклад, асистент говорить надто швидко, використовує складні речення або візуальна інформація суперечить голосовій). Мета розробника – мінімізувати цей тип навантаження.
3. **Релевантне (Germane):** зусилля, спрямовані на обробку інформації та навчання роботи з системою.

Практична рекомендація: не змушуйте користувача запам'ятовувати список опцій, озвучених голосом. Якщо варіантів більше трьох (наприклад, список знайдених книг), їх необхідно дублювати на екрані у вигляді тексту або карток. Голос має використовуватися для навігації та коротких підказок, а екран – для детальної інформації.

4.1.2. Ефект розділеної уваги

Ефект розділеної уваги (Split-attention effect) виникає, коли користувач змушений розподіляти увагу між різними джерелами інформації, які є незрозумілими одне без одного. У контексті курсової роботи це означає, що 3D-персонаж не повинен відволікати користувача, коли той читає опис книги або заповнює форму замовлення.

Стратегії уникнення конфлікту уваги

- **Синхронізація:** анімація персонажа має чітко відповідати моменту мовлення.
- **Фокусування:** коли система виводить результати пошуку, персонаж повинен переходити у стан спокою (Idle) або вказувати жестом на контент (Deictic gesture), перемикаючи увагу користувача на дані.

4.2. Особливості проєктування антропоморфних інтерфейсів

Антропоморфізм — це наділення людськими якостями (мовою, емоціями, зовнішністю) нелюдських об'єктів. Використання 3D-персонажа (втіленого агента) значно підвищує довіру до системи та залученість користувача, але приховує в собі ризики.

4.2.1. Феномен «Зловісної долини»

Однією з головних проблем при створенні 3D-асистентів є ефект «Зловісної долини» (Uncanny Valley), описаний японським робототехніком Масахіро Морі. Суть ефекту полягає в тому, що зі зростанням схожості робота або 3D-моделі на людину, симпатія користувачів зростає до певної межі. Однак, коли об'єкт стає *майже* як людина, але має дрібні недоліки

(неприродна міміка, "скляний" погляд, розсинхронізація губ), симпатія різко змінюється на відразу та страх.

Рекомендації для курсового проєктування: оскільки створення фотореалістичної анімації у браузері є технічно складним завданням, студентам наполегливо рекомендується використовувати **стилізованих персонажів** (Low-poly, Cartoon style, абстрактні роботи або тварини). Стилізовані персонажі не потрапляють у «зловісну долину», оскільки мозок користувача не очікує від них повної реалістичності, прощаючи спрощену анімацію.

4.2.2. Розроблення персони (Persona Design)

Персона – це сукупність характеристик, що формують "особистість" асистента. Без чітко визначеної персони діалог виглядатиме механічним та непослідовним. При розробленні персонажа необхідно визначити наступне.

- **Роль:** хто він? (бібліотекар, друг, експерт, помічник).
- **Тон голосу:** формальний, жартівливий, лаконічний чи емпатичний.
- **Ім'я та візуальний стиль:** вони мають відповідати призначенню системи.

4.2.3. Етичні аспекти

При проєктуванні асистента слід уникати гендерних стереотипів (наприклад, створення виключно жіночих персонажів для сервісних ролей). Також важливо запрограмувати поведінку асистента у випадку некоректних або образливих звертань з боку користувача — система має відповідати нейтрально та ввічливо, повертаючи діалог у конструктивне русло ("Вибачте, я не можу відповісти на це. Давайте повернемося до пошуку книг").

4.3. Принципи побудови людино-машинного діалогу

Проеєктування голосового інтерфейсу (VUI – Voice User Interface) кардинально відрізняється від графічного (GUI). Головна відмінність –

тимчасовість. Графічні елементи постійно присутні на екрані, тоді як голосові команди зникають миттєво після озвучення.

Максими Грайса в контексті VUI

Для створення природного діалогу слід спиратися на принципи кооперації Пола Грайса (Grice's Maxims), які полягають у наступному.

1. **Максима кількості (Quantity):** інформації має бути не менше і не більше, ніж потрібно.
 - *Погано:* "Я знайшов книгу 'Кобзар'. Вона написана Тарасом Шевченком, видана у 2010 році видавництвом 'А-ба-ба-га-ла-ма-га', має 300 сторінок, тверду обкладинку..." (Забагато інформації на слух).
 - *Добре:* "Я знайшов 'Кобзар' Тараса Шевченка. Бажаєте замовити чи дізнатися деталі?"
2. **Максима якості (Quality):** асистент повинен говорити правду. Якщо впевненість у розпізнаванні низька (Confidence Score < 70%), краще перепитати, ніж виконати хибну дію.
3. **Максима релевантності (Relation):** не відхилятися від теми запиту.
4. **Максима способу (Manner):** уникати двозначності, складних термінів та професійного жаргону.

Стратегії оброблення помилок

Помилки у голосових інтерфейсах неминучі (фоновий шум, акцент, невідомі слова). Якісна система відрізняється вмінням "відновлюватися" після помилок (Repair strategies).

Розрізняють наступні типи помилок та реакції системи.

- **Не розпізнано (No Match):** система почула звук, але не змогла зіставити його з жодним відомим наміром.
 - *Реакція:* "Вибачте, я не зрозумів. Ви хотіли знайти книгу чи перевірити статус замовлення?" (Пропозиція варіантів).
- **Тиша (No Input):** мікрофон був увімкнений, але користувач мовчав.

- *Реакція:* "Я все ще слухаю. Натисніть кнопку мікрофона, коли будете готові говорити."
- ***Помилка бекенду (System Error):***
 - *Реакція:* "На жаль, сервер зараз не відповідає. Спробуйте пізніше."

Мультимодальний зворотний зв'язок

Критично важливим для VUI є підтвердження стану системи. Користувач завжди має розуміти, чи слухає його асистент. Рекомендується використовувати трирівневу індикацію.

1. ***Візуальна (3D-персонаж):*** анімація очікування, нахил голови/вуха, візуалізація звукової хвилі.
2. ***Аудіальна (Earcons):*** короткі звуки (ping/pop) при активації та деактивації мікрофона.
3. ***Текстова:*** дублювання розпізнаної фрази на екрані (субтитри), щоб користувач міг переконатися у правильності сприйняття команди.

5. МЕТОДИКА ПРОЄКТУВАННЯ ГОЛОСОВОЇ ВЗАЄМОДІЇ

Розроблення голосового інтерфейсу починається не з написання коду, а з моделювання діалогу. На відміну від графічного інтерфейсу, де користувач обмежений кнопками, у голосовому інтерфейсі він може сказати що завгодно. Мета цього етапу – структурувати цю невизначеність.

5.1. Сценарне проєктування (Scripting)

Першим кроком є написання зразків діалогів (Sample Dialogs). Це текстовий опис ідеальної взаємодії між користувачем та системою, схожий на кіносценарій.

Структура зразка діалогу

Діалог записується у вигляді таблиці з двома колонками: «Користувач» та «Асистент». Обов'язково додавайте колонку «Контекст/Дія» для опису того, що відбувається на екрані або що робить 3D-персонаж.

Приклад сценарію «Пошук книги» наведено у табл. 5.1.

Таблиця 5.1

Хто говорить	Репліка	Контекст / Дія системи
Користувач	Привіт! Знайди мені щось почитати з фантастики.	Натискає кнопку мікрофона.
Асистент	Вітаю! Шукаю фантастику. Уточність, будь ласка: наукова фантастика чи фентезі?	Анімація: Speaking. Екран: Виводить категорії.
Користувач	Наукова. Бажано про космос.	Анімація: Listening.
Асистент	Зрозуміло. Ось найкращі книги про космос. Рекомендую «Дюну» або «Марсіанина».	Анімація: Pointing (вказує крилом на список). Екран: Список книг з обкладинками.

Проектування «нещасливих шляхів»

Студенти часто припускаються помилки, проєктуючи лише успішний сценарій (Happy Path – щасливий шлях). Необхідно також прописати сценарії для помилок.

1. **Не розпізнано (No Match):** користувач сказав нісенітницю.

- *Асистент:* «Вибачте, я не розчув назву жанру. Повторіть, будь ласка».

2. **Немає результатів (Empty State):** книгу не знайдено в БД.

- *Асистент:* «На жаль, у нас немає книг цього автора. Можливо, вас зацікавлять схожі жанри?»

5.2. Робота з намірами (Intents) та сутностями (Slots)

Для того щоб програмна система зрозуміла людську мову, вхідну фразу необхідно розкласти на структуровані дані. Цей процес називається NLU (Natural Language Understanding).

Наміри (Intents)

Intent – це мета користувача, те, *що* він хоче зробити. Вся фраза класифікується як один намір.

Приклади намірів для бібліотечної системи є наступними.

- Greeting: «Привіт», «Добрий день», «Hello».
- SearchBook: «Знайди книгу», «Шукаю роман», «Покажи підручники».
- BookOrder: «Замовити», «Хочу це прочитати», «Бронюю».
- Help: «Що ти вмієш?», «Допоможи».

Сутності або Слоти (Entities / Slots)

Slot – це деталі або параметри запиту, з *чим* користувач хоче виконати дію. Це змінні, які система має «витягнути» з фрази.

Приклад розбору фрази: *"Знайди книги Стівена Кінга англійською мовою"*.

Система має розпізнати:

- **Intent:** SearchBook
- **Slot (Author):** Stephen King
- **Slot (Language):** English

Під час курсового проєктування складіть таблицю намірів та слотів для свого проєкту. Це стане основою для конфігурації хмарних сервісів (Dialogflow, Wit.ai) або написання власної логіки пошуку за ключовими словами.

5.3. Візуалізація логіки діалогу

Після написання скриптів необхідно створити **блок-схему (Flowchart)** алгоритму роботи асистента. Це обов'язкова частина документації.

Основні елементи схеми є наступними.

- **Початок (Start):** активація мікрофона (Wake word або клік).
- **Дія (Process):** розпізнавання мови (STT).
- **Рішення (Decision - Ромб):** перевірка Intent.
 - *Якщо Intent = Search:* перехід до пошуку в БД.
 - *Якщо Intent = Unknown:* перехід до гілки обробки помилок.
- **Вивід (Output):** синтез мовлення (TTS) та анімація відповіді.

Схема повинна покривати замкнений цикл: від запиту користувача до очікування наступної команди.

5.4. Формування навчальної вибірки

Навіть якщо ви використовуєте просте порівняння рядків, система повинна знати синоніми. Користувачі можуть сформулювати один і той самий запит по-різному.

Прикладом словника синонімів (Utterances) може слугувати наступне.

- **Запит:** «Фантастика»

- **Синоніми:** «Sci-Fi», «Сайнс фікшн», «Про майбутнє», «Фентезі» (якщо категорії об'єднані).

Для курсового проєкту рекомендується підготувати мінімум 5-10 варіантів фраз для кожного ключового наміру.

5.5. Мультимодальна синхронізація

Стани голосового інтерфейсу мають бути чітко прив'язані до станів 3D-моделі та графічного інтерфейсу (GUI): табл. 5.2.

Таблиця 5.2

Таблиця станів системи

Стан системи (VUI)	Анімація персонажа (3D)	Графічний інтерфейс (GUI)
Очікування (Idle)	Легкий рух, кліпання очима (Idle Loop).	Кнопка мікрофона активна.
Слухання (Listening)	Нахил корпусу вперед, рука біля вуха.	Індикація запису (пульсація), кнопка «Стоп».
Оброблення (Processing)	Задумливий вигляд (чухає потилицю).	Спінер завантаження (Loader).
Відповідь (Speaking)	Активна артикуляція (Lip-sync), жести руками.	Субтитри відповіді, картки результатів.

Реалізація цієї таблиці у коді забезпечує цілісність сприйняття асистента як живої істоти.

6. ТЕХНОЛОГІЇ 3D-ВІЗУАЛІЗАЦІЇ ТА АНІМАЦІЇ У ВЕБ-СЕРЕДОВИЩІ

Візуалізація мультимедійного помічника є ключовим елементом курсового проєкту. На відміну від статичних зображень, 3D-асистент повинен реагувати на дії користувача в реальному часі. Це вимагає використання специфічних форматів даних та алгоритмів анімації, оптимізованих для роботи у веб-браузері.

6.1. Створення та оптимізація 3D-моделей для Web

Головним обмеженням веб-графіки є продуктивність пристрою користувача та швидкість інтернет-з'єднання. Тому модель персонажа має бути максимально легкою («Low-poly»).

Вимоги до топології

Топологія – це структура полігональної сітки моделі. Для веб-анімації існують наступні правила.

1. **Полігонаж (Polycount):** рекомендована кількість трикутників для одного персонажа у браузері — до **10-20 тисяч**. Використання високополігональних моделей (High-poly) призведе до падіння FPS (кадрів у секунду) та зависання сторінки.
2. **Квадратна сітка (Quads):** для анімації, особливо обличчя, бажано використовувати чотирикутні полігони. Вони краще деформуються при згинанні, ніж трикутники, уникаючи візуальних артефактів на "суглобах" моделі.
3. **Edge Loops:** навколо рота та очей сітка повинна йти концентричними колами. Це критично важливо для коректної міміки під час розмови.

Формати файлів: glTF/GLB

Стандартом де-факто для передачі 3D-сцен у вебі є формат **glTF (GL Transmission Format)**, який часто називають «JPEG для 3D».

Переваги glTF

- Компактність: файл містить геометрію, текстури та анімацію у стиснутому вигляді.
- Швидкість завантаження: структура файлу готова до прямого завантаження у відеопам'ять (GPU) без складного парсингу.
- Підтримка PBR (Physically Based Rendering): матеріали виглядають реалістично при різному освітленні.

GLB vs glTF: Для курсового проєкту рекомендується використовувати бінарну версію — **.glb** (один файл), оскільки це спрощує завантаження моделі на сервер.

6.2. Анімація персонажа

Анімація у вебі реалізується двома основними методами: скелетною анімацією (для тіла) та морфінгом (для обличчя).

Скелетна анімація (Rigging & Skinning)

Для того щоб модель рухалася, всередину неї поміщається ієрархічна структура «кісток» (Rig).

- **Скінінг (Skinning):** процес прив'язки вершин сітки до кісток. Кожна вершина має «вагу» впливу (Weight Paint). Наприклад, вершини ліктя повинні рухатися на 50% за плечем і на 50% за передпліччям для плавного згину.
- **Цикли анімації:** в редакторі (Blender/Maya) створюються окремі кліпи:
 - *Idle:* циклічне дихання, легке погойдування.
 - *Wave/Greet:* привітання рукою.
 - *Gestures:* вказівні жести для акцентування уваги на інтерфейсі.

Морфінг (Blend Shapes / Shape Keys)

Скелетна анімація погано підходить для складної міміки. Для анімації обличчя використовується технологія **Blend Shapes** (або Shape Keys у

Blender). Це набір копій сітки обличчя, де кожна копія має змінену форму (наприклад, піднята брова, заплющене око, відкритий рот). У кодї веб-додатку ми можемо плавно змішувати ці форми, встановлюючи значення від 0 до 1 (наприклад, `mouthOpen = 0.7`).

6.3. Автоматичний ліпсінк (Lip Sync)

Синхронізація руху губ зі звуком є обов'язковою вимогою до мультимедійного асистента.

Поняття візем (Visemes)

Візема — це візуальне відображення фонем (звуку). Оскільки багато звуків виглядають однаково (наприклад, [b], [m], [p]), кількість візем менша за кількість фонем. Для реалістичної анімації мовлення необхідно створити Shape Keys для основних візем.

1. **A / I:** широко відкритий рот.
2. **O / U:** губи витягнуті у трубочку.
3. **M / B / P:** губи стиснуті.
4. **F / V:** нижня губа торкається верхніх зубів.
5. **Rest:** нейтральне положення (рот закритий).

Алгоритмічна реалізація

Існує два підходи до ліпсінку в браузері:

- 1) **амплітудний (простий)** – відкриття рота прив'язується до гучності звуку. Чим голосніший звук, тим ширше відкривається рот (змінюється вага Shape Key "OpenMouth"); це виглядає спрощено, як у ляльок-маппетів;
- 2) **фонемний (просунутий)** – використання Web Speech API або серверних бібліотек (наприклад, *Oculus Lipsync*) для розпізнавання фонем у реальному часі; отримані фонемні мапляються на відповідні віземи 3D-моделі з плавним переходом (інтерполяцією) між ними.

6.4. Програмна інтеграція (Three.js)

Для відображення 3D-сцени у браузері використовується бібліотека **Three.js**, яка є надбудовою над WebGL.

Базова архітектура сцени

Типовий код ініціалізації 3D-асистента включає:

- **Scene:** контейнер для всіх об'єктів;
- **Camera:** "очі" користувача (зазвичай `PerspectiveCamera`), Для портретного відображення асистента фокусна відстань налаштовується на рівні 50-85mm;
- **Renderer:** компонент, що малює сцену на `<canvas>` елементі HTML;
- **Light:** освітлення (`AmbientLight` для загального фону та `DirectionalLight` як основне джерело, щоб створити об'єм).

Завантаження та керування анімацією

Завантаження моделі виконується через `GLTFLoader`. Для програвання анімацій використовується `AnimationMixer`. Ключовим моментом є перемикання станів:

```
// Псевдокод перемикання анімації (JavaScript)
function playAction(name) {
    const action = mixer.clipAction(animations[name]);
    action.reset();
    action.fadeIn(0.5); // Плавний перехід за 0.5 сек
    action.play();
}
```

Такий підхід дозволяє асистенту плавно переходити зі стану "Спокій" у стан "Слухання" без різких ривків моделі.

7. ПРОГРАМНА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

Цей етап є кульмінацією роботи над курсовим проектом. На основі розробленої архітектури, сценаріїв діалогу та 3D-моделі необхідно створити працюючий програмний продукт. У цьому розділі розглядаються особливості реалізації клієнтської та серверної частин, а також їх інтеграція.

7.1. Архітектура системи

Для мультимедійних систем із голосовим управлінням рекомендується використовувати класичну клієнт-серверну архітектуру (рис. 7.1), яка забезпечує гнучкість та масштабованість.

Загальна схема взаємодії

Система складається з трьох логічних рівнів.

1. **Клієнт (Frontend):** односторінковий веб-додаток (SPA), що відповідає за візуалізацію (3D-сцена), захоплення аудіо та відтворення відповідей.
2. **Сервер додатка (Backend):** обробляє бізнес-логіку (пошук у базі даних, управління користувачами).
3. **Хмарні сервіси (Cloud AI):** зовнішні API для ресурсомістких задач розпізнавання (STT) та синтезу (TTS) мовлення.

Потік даних (Data Flow)

Мікрофон користувача $\rightarrow$ Аудіопотік $\rightarrow$ STT (Текст) $\rightarrow$ Backend (Пошук Intent) $\rightarrow$ Дія в БД $\rightarrow$ Текстова відповідь $\rightarrow$ TTS (Аудіо) + Анімація $\rightarrow$ Клієнт.

7.2. Реалізація клієнтської частини (Frontend)

Клієнтська частина є точкою входу для користувача. Основні задачі: рендеринг інтерфейсу та управління станом асистента.

Технологічний стек

- **HTML5/CSS3:** базова розмітка та стилізація.
- **JavaScript (ES6+) / TypeScript:** мова програмування логіки. Рекомендується TypeScript для суворої типізації даних (інтерфейсів відповідей API).
- **React / Vue.js:** фреймворки для створення реактивного інтерфейсу. Вони дозволяють миттєво оновлювати стан UI (наприклад, показувати спінер завантаження), не перезавантажуючи сторінку.
- **Three.js / React-Three-Fiber:** бібліотеки для відображення 3D-асистента.

Робота з Web Audio API

Для запису голосу використовується браузерний MediaRecorder API.

Важливий нюанс: браузери блокують доступ до мікрофона, якщо сайт не використовує захищений протокол HTTPS (на локальному сервері localhost це обмеження зазвичай не діє).

Управління станом (State Management)

Необхідно реалізувати машину станів (State Machine) для асистента, щоб уникнути логічних помилок (наприклад, асистент не повинен слухати, поки сам говорить).

- *States:* IDLE, LISTENING, PROCESSING, SPEAKING, ERROR.

7.3. Реалізація серверної частини (Backend)

Серверна частина відповідає за логіку діалогу та роботу з даними.

Вибір технології

- **Node.js (Express/NestJS):** ідеально підходить для роботи з JSON та WebSocket завдяки асинхронній природі.
- **Python (Flask/FastAPI):** рекомендується, якщо планується використання складних бібліотек NLP (NLTK, SpaCy) або інтеграція

з ML-моделями, оскільки Python є стандартом у сфері штучного інтелекту.

Оброблення намірів (Intent Matching)

Якщо не використовуються готові платформи (як Dialogflow), на бекенді необхідно реалізувати алгоритм пошуку ключових слів.

- *Простий підхід*: пошук підстроки (if "знайди" in text).
- *Гнучкий підхід*: використання відстані Левенштейна (Fuzzy Matching) для компенсації помилок розпізнавання (наприклад, "знайти" і "знайди").

7.4. Інтеграція з хмарними сервісами (Cloud Integration)

Високоякісне розпізнавання української мови наразі найкраще реалізовано у хмарних провайдерів.

Налаштування Google Cloud Platform (GCP)

Для використання Google Cloud Speech-to-Text та Text-to-Speech необхідно виконати наступні кроки.

1. Створити проєкт у Google Cloud Console.
2. Активувати відповідні API.
3. Отримати ключ доступу (API Key) або файл сервісного акаунту (credentials.json).
4. **Увага:** ключі не можна зберігати у клієнтському коді (frontend), оскільки зловмисники можуть їх викрасти. Запити до Google API мають йти **через ваш бекенд** (Proxying requests).

Оптимізація затримок (Latency)

Голосовий інтерфейс має бути швидким. Затримка більше 1-2 секунд руйнує ілюзію живого діалогу.

- *Рекомендація*: використовуйте потокову передачу (Streaming) для STT, щоб розпізнавання йшло паралельно з мовленням користувача, а не після завершення запису файлу.

7.5. Тестування системи

Перевірка працездатності мультимедійної системи включає специфічні види тестування.

Тестування VUI (Voice User Interface)

- **Тест на акценти та шуми:** перевірка розпізнавання у тихому приміщенні та з фоновим шумом (кафе, вулиця).
- **Тест сценаріїв:** проходження всіх гілок діалогу, включаючи тупикові (коли система не розуміє запит 3 рази поспіль).

7.5.2. Тестування продуктивності

- **FPS (Frames Per Second):** 3D-анімація не повинна «просідати» нижче 30 кадрів/сек на цільових пристроях.
- **Memory Leak:** перевірка витоків пам'яті при тривалому використанні (чи не "зависає" браузер після 50 голосових команд).

ДОДАТКИ

ДОДАТОК А. Приклад титульного аркуша

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
„Київський політехнічний інститут імені Ігоря Сікорського”
Факультет програмних систем та прикладної математики
Кафедра програмного забезпечення комп’ютерних систем**

КУРСОВИЙ ПРОЄКТ

з дисципліни “Мультимедійні інтерфейси та 3D візуалізація”

на тему:

**ІНТЕРАКТИВНА AR-СИСТЕМА ФОРМУВАННЯ ЗАМОВЛЕНЬ У
РЕСТОРАНАХ ШВИДКОГО ХАРЧУВАННЯ ЗА ДОПОМОГОЮ
ГОЛОСОВОГО УПРАВЛІННЯ. МОДУЛЬ ЗБОРУ ТА ОБРОБКИ
ДАНИХ**

Виконав студент групи КП-11мп
Іваненко Іван Іванович
Залікова книжка КП-1104

Керівник проєкту: д.т.н. Сулема Є.С.

До захисту допущено

(дата, підпис)

Захищено з оцінкою

(дата, підпис)

Київ – 2026

ДОДАТОК Б. Приклад технічного завдання

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ „КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”

**Факультет програмних систем та прикладної математики
Кафедра програмного забезпечення комп’ютерних систем**

“ЗАТВЕРДЖЕНО”

Керівник проекту

_____ Євгенія СУЛЕМА

“ ____ ” _____ 2026 р.

ІНТЕРАКТИВНА СИСТЕМА РОБОТИ КІНОТЕАТРУ З ВИКОРИСТАННЯМ МУЛЬТИМЕДІЙНОГО ПОМІЧНИКА ТА ГОЛОСОВОГО АСИСТЕНТА

Технічне завдання

ПЗКС.045440-02-91

Виконавці:

_____ Іван ІВАНЕНКО

_____ Марія МАРІЙЧЕНКО

_____ Сергій СЕРГІЄНКО

ЗМІСТ

1. Найменування та галузь застосування	3
2. Підстава для розроблення	3
3. Призначення розробки	3
4. Вимоги до програмного продукту	3
5. Вимоги до проєктної документації	4
6. Етапи проєктування	4

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: інтерактивна система роботи кінотеатру з використанням мультимедійного помічника та голосового асистента.

Галузь застосування: інформаційні технології.

2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ

Підставою для розроблення є завдання на курсовий проєкт з дисципліни «Мультимедійні інтерфейси та 3D-візуалізація».

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для використання зацікавленими користувачами з метою організації роботи кінотеатру, представлення актуальної інформації щодо сеансів та фільмів, з можливістю бронювання місць в залі, використовуючи графічний інтерфейс та систему голосового управління

4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

Додаток повинен забезпечувати такі основні функції.

1. Користувач може запитувати у мультимедійного помічника інформацію про фільми.
2. Користувач може запитувати у мультимедійного помічника актуальну інформацію про сеанси.
3. Користувач може запитувати у мультимедійного помічника актуальну інформацію про вільні місця в залі.
4. Система повинна надавати можливості реєстрації та авторизації.
5. Користувач може забронювати та придбати квитки за допомогою голосового асистента.

6. Користувач може відмінити бронювання та повернути придбаний квиток за допомогою голосового асистента.

5. ВИМОГИ ДО ПРОЄКТНОЇ ДОКУМЕНТАЦІЇ

У процесі виконання проєкту повинна бути розроблена наступна документація:

- 1) пояснювальна записка;
- 2) керівництво користувача.

6. ЕТАПИ ПРОЄКТУВАННЯ

Вивчення літератури за тематикою проєкту	08.09.2026
Розроблення та узгодження технічного завдання	27.09.2026
Розроблення структури застосунку.....	15.10.2026
Розроблення дизайну додатку та графічних елементів	25.10.2026
Програмна реалізація голосового асистента	01.11.2026
Представлення бета-версії застосунку	27.11.2026
Оформлення технічної документації проєкту	05.12.2026

ДОДАТОК В. Приклад пояснювальної записки

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
„Київський політехнічний інститут імені Ігоря Сікорського”
Факультет програмних систем та прикладної математики
Кафедра програмного забезпечення комп’ютерних систем**

“ЗАТВЕРДЖЕНО”

Керівник проекту

_____ Євгенія СУЛЕМА

“ ____ ” _____ 2026 р.

**ІНТЕРАКТИВНА СИСТЕМА ПОШУКУ КНИГ У БІБЛІОТЕЦІ З
ВИКОРИСТАННЯМ МУЛЬТИМЕДІЙНОГО ПОМІЧНИКА ТА
ГОЛОСОВОГО АСИСТЕНТА. АРХІТЕКТУРА І ЛОГІКА
ПРОГРАМНОЇ СИСТЕМИ ТА ІНТЕГРАЦІЯ КОМПОНЕНТ
ПРОГРАМНОГО ПРОДУКТУ**

Пояснювальна записка

Виконавець:

_____ Андрій АНДРІЙЧЕНКО

2026

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ	3
ВСТУП	4
1. АНАЛІЗ ІНТСРУМЕНТІВ ПРОГРАМУВАННЯ ТА ТЕХНОЛОГІЙ РОЗРОБЛЕННЯ СЕРВЕРНОЇ ЧАСТИНИ	6
1.1. Мова програмування Python.....	6
1.2. Мова програмування TypeScript	9
1.3. Аналіз використаних веб-фреймворків	11
1.4. Аналіз використаних бібліотек	13
2. РОЗРОБЛЕННЯ АРХІТЕКТУРИ І ЛОГІКИ ПРОГРАМИ.....	16
2.1. Аналіз вимог до розроблюваного програмного забезпечення.....	16
2.2. Опис вимог до інтерактивної системи пошуку книг	18
2.3. Архітектура системи	20
2.4. Структура серверної частини	22
3. АНАЛІЗ РОЗРОБЛЕНИХ МОДУЛІВ.....	24
3.1. Особливості реалізації модулю STT.....	24
3.2. Особливості реалізації модулю TTS.....	27
3.3. Тестування розроблених компонентів	28
3.4. Рекомендації щодо подальшого вдосконалення	29
ВИСНОВКИ	30
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ	31

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

HTTP (скор. від англ. HyperText Transfer Protocol) – протокол передачі даних, що використовується в комп'ютерних мережах. Використовується для передачі гіпертекстових документів.

Інтернет – всесвітня система сполучених комп'ютерних мереж по всьому світу, яка використовує стек протоколів TCP/IP для здійснення комунікації між її учасниками.

ПЗ – програмне забезпечення, що складається з сукупності програм для виконання комп'ютером.

Машинне навчання – алгоритми із галузі штучного інтелекту, для яких характерне не пряме рішення поставленої задачі, а прогнозування результату після проходження процесу навчання на зібраних даних, використовуючи при цьому математичну статистику, теорію ймовірностей тощо.

API (скор. від англ. application programming interface) – прикладний програмний інтерфейс, який надає набір чітко визначених правил для взаємодії одних програм з іншими.

3D (скор. від англ. 3 Dimensions) – розділ комп'ютерної графіки, сукупність прийомів та інструментів (як програмних, так і апаратних), призначених для зображення об'ємних об'єктів.

STT (скор. від англ. Speech-to-Text) – технологія перетворення голосу в текстову інформацію.

TTS (скор. від англ. Text-to-Speech) – технологія перетворення текстової інформації в аудіо файл, в якому озвучується людським голосом надані слова у вигляді тексту.

ВСТУП

На сьогоднішній день існує достатньо програмного забезпечення, яке застосовується в різних сферах життя. Очевидним є той факт, що розробники та дизайнери прагнуть створити такий застосунок, який би був найбільш зручним для використання кінцевими користувачами, тому що в світі загалом існує досить багато програмного забезпечення майже для всіх можливих задач, але користуються попитом лише ті програми, які виконують свої функціональні можливості за мінімальну кількість взаємодій, і причому коли ці взаємодії є інтуїтивними та зрозумілими. Тому в останні роки можна побачити стрімкий розвиток голосових асистентів, за допомогою яких можна швидко виконати будь-які дії з однієї «точки входу». А якщо даний асистент є мультимедійним, тобто має певне візуальне представлення, то даний продукт може очікувати на значну популярність серед користувачів.

.....

Цей проєкт присвячений розробленню архітектури і логіки програмної системи та інтеграції компонент програмного продукту, який призначений для спрощення пошуку та замовлення книг для відвідувачів бібліотек і книгарень, а також для демонстрування можливостей сучасних мультимедійних технологій.

1. АНАЛІЗ ІНСТРУМЕНТІВ ПРОГРАМУВАННЯ ТА ТЕХНОЛОГІЙ РОЗРОБЛЕННЯ СЕРВЕРНОЇ ЧАСТИНИ

1.1. Мова програмування Python

Мова програмування Python за останні десятиріччя де-факто стала стандартом проведення обчислень для багатьох наукових досліджень. Це і не дивно, оскільки популярність даної мови протягом останніх років тільки зростає [1] і зараз вона посідає передову позицію у рейтингу за кількістю запитів про неї [2]. Python є інтерпретованою мультипарадигмальною мовою програмування з динамічною типізацією, що дозволяє розв'язувати задачі для різних предметних галузей.

Мова програмування Python має наступні переваги:

- динамічна типізація з підтримкою багатьох парадигм програмування, що надає швидкість прототипування та гнучкість під час розроблення ПЗ;
- наявність засобів для асинхронного виконання функцій та розпаралелювання задач за допомогою потоків;

.....

1.2. Мова програмування TypeScript

Мова програмування TypeScript – це мова з відкритим кодом, яка базується на JavaScript, одному з найбільш використовуваних інструментів у світі, додаючи статичні визначення типів. Типи — це спосіб описати форму об'єкта, надаючи кращу документацію і даючи можливість TypeScript перевірити правильність роботи коду. Типи написання можуть бути не обов'язковими для TypeScript, оскільки умова типу дозволяє отримати багато можливостей без написання додаткового коду [4]. Весь дійсний код JavaScript також є кодом TypeScript.

.....

1.3. Аналіз використаних веб-фреймворків

1.3.1. Аналіз веб-фреймворку Flask

Flask – один із найпопулярніших мікрофреймворків для швидкого створення веб-додатків на мові Python. Хоч і в ньому досить небагато вбудованих компонентів, але це компенсується можливістю підключення сторонніх розширень [42].

.....

1.3.2. Аналіз веб-фреймворку Express

За фреймворк було обрано Express, оскільки надає розробнику можливість бути вільним у виборі засобів вирішення тієї чи іншої проблеми.

.....

1.4. Аналіз використаних бібліотек

1.4.1. Аналіз використаних бібліотек для роботи з аудіо

В основному для роботи з аудіо було використано бібліотеку pydub, написану на мові програмування Python. Ця бібліотека дозволяє керувати аудіо за допомогою простого та легкого інтерфейсу високого рівня.

.....

1.4.2. Аналіз використаних бібліотек для роботи з 3D моделлю

Для відображення 3D моделі на веб-сторінці, були використана бібліотека Three.js [7], а також інші супутні бібліотеки для інтеграції в фреймворк React та відображення анімацій.

.....

2. РОЗРОБЛЕННЯ АРХІТЕКТУРИ І ЛОГІКИ ПРОГРАМИ

2.1. Аналіз вимог до розроблюваного програмного забезпечення

Однією з складових частин процесу розробки програмного забезпечення є аналіз вимог. Він включає в себе збір вимог до програмного забезпечення, їх систематизацію, виявлення взаємозв'язків, а також подальше документування. Під час збору вимог важливо приділяти увагу можливим протиріччям вимог у різних зацікавлених осіб, наприклад, у замовників та розробників.

.....

2.2. Опис вимог до інтерактивної системи пошуку книг з асистентом

Зібрані вимоги були розподілені між розробниками та кожній з них призначено ідентифікатор – код вимоги.

.....

2.3. Архітектура системи

Загалом розроблювану систему можна описати клієнт-серверною взаємодією. Тобто маємо серверну частину, в якій виконуються основні операції над даними, та клієнтську частину, яка відповідає за графічну та голосову взаємодію з користувачем.

Дана система складається з таких модулів:

- модуль серверної частини:
 - модуль взаємодії з сутностями системи;
 - модуль обробки аудіо даних;
- модуль клієнтського інтерфейсу:
 - модуль інтерактивної взаємодії через голосовий асистент;
 - модуль графічної взаємодії через елементи веб-сторінки.

Діаграма архітектури системи відображена на рис. 2.1.

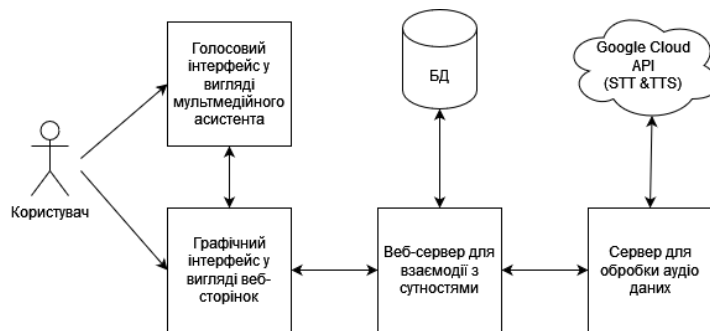


Рис. 2.1. Архітектура розроблюваної системи

Як вже було сказано раніше, модуль серверної частини складається з двох частин. Веб-сервер для взаємодії з сутностями відповідає за проведення операцій з базою даних та виконання операцій на основі цих даних (реєстрації, авторизації, створення, видалення, отримання та редагування сутностей). В свою чергу, сервер для обробки аудіо даних відповідає за основні операції взаємодії за допомогою голосу, а саме перетворення звуку у текст та голосовий синтез заданого тексту.

Модуль клієнтського інтерфейсу представляє собою односторінковий веб-сайт, який дозволяє користувачу виконувати основні операції для бібліотеки. Складається з двох частин: графічна частина відповідає за відображення сторінок, статичних та динамічних даних на них та взаємодію з частиною голосового вводу. Голосовий інтерфейс у свою чергу відповідає за інтерактивну взаємодію з користувачем за допомогою голосу та виконання операцій пошуку книжок, а також інших дій, як-от: перехід на певний розділ сайту або на сторінку книжки.

2.4. Структура серверної частини

Серверна частина для взаємодії з сутностями бібліотек складається з наступних частин:

- моделі сутностей;
- контролери сутностей;

.....

3. АНАЛІЗ РОЗРОБЛЕНИХ МОДУЛІВ. Особливості реалізації модулю STT

Перед тим як оброблювати будь-який аудіо файл, необхідно було конвертувати у формат wav з потрібними параметрами частоти дискретизації тощо. Для цього було застосовано бібліотеку pydub, а саме застосовано об'єкт AudioSegment.

.....

3.2. Особливості реалізації модулю TTS

Для перетворення текстової інформації в голос також було застосовано Google Cloud Text-To-Speech API. Щоби скористатись відповідною сторонньою функціональністю було використано бібліотеку gTTS на мові програмування Python.

.....

3.3. Тестування розроблених компонентів

Тестування системи відбулося шляхом мануального тестування.

Для тестування було написано та протестовані такі варіанти:

- перетворення стандартних голосових команд в текст;
- перетворення тексту в голос;

.....

3.4. Рекомендації щодо подальшого вдосконалення

В процесі роботи над інтерактивною системою пошуку книг у бібліотеці з використанням мультимедійного помічника та голосового асистента було виявлено ряд напрямів та функцій, які можуть покращити розроблене ПЗ, а саме:

- застосувати технологію Electron для клієнтської частини програми, щоби мати змогу сформувати виконуваний файл для різних платформ, навіть для тих, що не мають веб-браузера;

.....

ВИСНОВКИ

Метою курсового проєкту було розроблення інтерактивної системи пошуку книг у бібліотеці з використанням мультимедійного помічника та голосового асистента. Зокрема в цій частині було розглянуто розроблення архітектури і логіки програмної системи та інтеграція компонент програмного продукту.

Аналіз засобів розроблення мультимедійних помічників, показав доцільність створення веб-серверів на мові програмування TypeScript та Python та застосування відкритих API для задач STT та TTS.

Розроблена інтерактивна система пошуку книг:

- може створювати профілі користувачів в системі;
- використовувати графічний інтерфейс для пошуку книг;
- використовувати голосовий інтерфейс для пошуку книг;
- замовляти книжки для подальшого читання;
- відображає 3D модель асистента, яка має анімації;
- реагує на ввід за допомогою анімацій та озвучення результатів обробки запитів..

Розробка виконана у повному обсязі, всі вимоги, викладені у технічному завданні враховані, тестування продукту виконано у відповідності до затвердженої програми та методики тестування.

Використання розробленої інтерактивної системи пошуку книг дозволить спростити пошук та замовлення обраної книги (або будь-якого іншого друкованого видання) для відвідувачів бібліотек і книгарень. Також голосовий асистент дозволить взаємодіяти з системою дуже швидко, що є особливо актуальним для літніх людей, яким зазвичай незручно використовувати клавіатуру. А мультимедійний помічник зробить цей процес більш інтерактивним, що може стати фактором популяризації читання книг серед дітей.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Millman, K.J. Python for Scientists and Engineers [Text] / K.J. Millman, M. Aivazis // Computing in Science & Engineering. — 2021. — Vol. 13, № 2. — P. 9 – 12.
2. Aslam, F.A. Efficient way of web development using python and flask [Text] / F.A. Aslam, H.N. Mohammed, P. Lokhande // International Journal of Advanced Research in Computer Science. — 2025. — Vol. 6, № 2. — P. 54 – 57.
3. TypeScript Programming. [Електронний ресурс]. — Режим доступу: <https://code.visualstudio.com/docs/languages/typescript>
4. Use TypeScript to Build a Node API with Express [Електронний ресурс]. Режим доступу: <https://developer.okta.com/blog/2018/11/15/node-express-typescript>
5. How Does Speech to Text Software Work? [Електронний ресурс]. — Режим доступу: <https://www.amberscript.com/en/blog/how-speech-to-text-software-works/>
6. Text-to-Speech Synthesis: An Overview. [Електронний ресурс]. — Режим доступу: <https://medium.com/sciforce/text-to-speech-synthesis-an-overview-641c18fcd35f>
7. An introduction to Three.js. [Електронний ресурс]. — Режим доступу: <https://humaan.com/blog/web-3d-graphics-using-three-js/>

ДОДАТОК 4. Приклад керівництва користувача

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
„Київський політехнічний інститут імені Ігоря Сікорського”
Факультет програмних систем та прикладної математики

“ЗАТВЕРДЖЕНО”

Керівник проекту

_____ Євгенія СУЛЕМА

“ ____ ” _____ 2026 р.

ІНТЕРАКТИВНА СИСТЕМА ПОШУКУ КНИГ У БІБЛІОТЕЦІ З
ВИКОРИСТАННЯМ МУЛЬТИМЕДІЙНОГО ПОМІЧНИКА ТА
ГОЛОСОВОГО АСИСТЕНТА

Керівництво користувача

ПЗКС.045440-05-34

Виконавець:

_____ Андрій АНДРІЙЧЕНКО

2026

ЗМІСТ

1. Опис структури системи	54
2. Опис пошуку книг	54
3. Опис реєстрації та авторизації	55
4. Опис створення та редагування замовлення	56
5. Опис створення та редагування книг	57

1. Опис структури системи

Система складається з наступних сторінок.

- Головна сторінка
- Сторінка логіна
- Сторінка авторизації
- Сторінка пошуку
- Сторінка перегляду деталей книги
- Сторінка перегляду замовлень
- Сторінки редагування та створення замовлень
- Сторінки редагування та створення книг
- Сторінки редагування та створення бібліотек

Кожна сторінка містить зверху хедер, де відображені посилання для переходу між сторінками.

2. Опис пошуку книг

Для пошуку книг користувачеві потрібно перейти на відповідну сторінку. По лівому краю сторінки розташовані фільтри по бібліотеках, авторах, видавництвах, типу книги, ключових словах, мові тексту та року друку. Окрім цього зверху вказане текстове поле для вводу частини назви книги.

По правому краю розташовані елементи голосового асистента: зверху 3D модель сови, а знизу кнопка з іконкою мікрофона для початку взаємодії з асистентом. По центру сторінки відображаються результати пошуку.

На рис. 1 зображено приклад пошуку. За допомогою голосового асистента був обраний фільтр за типом книги, а саме роман. Після цього був виконаний пошук за даним фільтром та відображені результати у таблиці.

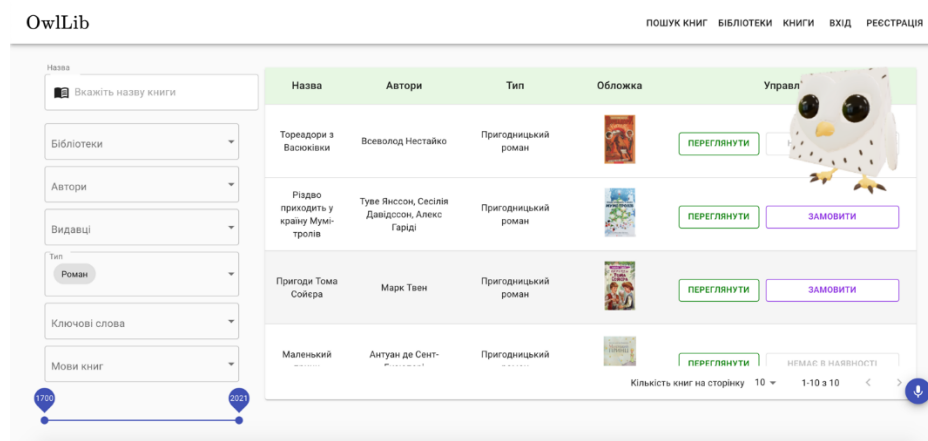


Рис. 1. Сторінка пошуку книг з голосовим асистентом

3. Опис реєстрації та авторизації

Використавши відповідні посилання в хедері сайту, користувач може перейти на сторінки реєстрації та авторизації, де буде відображатися відповідна форма (рис. 2).

Рис. 2. Сторінка реєстрації

Після заповнення форми коректними даними користувач може перейти на сторінку акаунта та своїх замовлень (рис. 3). При некоректних даних відповідні повідомлення з помилка будуть відображені, а поля пофарбуються у червоний колір.

Усі замовлення в системі										
Номер замовлення	Спосіб видачі	Читач	Книжка	Бібліотека	Дата створення	Готовність до видачі	Дедлайн повернення	Дата отримання	Дата повернення	Управління
1	У бібліотеці	Doe John	Найкоротша історія часу	Бібліотека ім. Лесі Українки	03.12.2021 13:00:23	Так	04.01.2022	13.12.2021	Ще не повернуто	ОНОВИТИ ВИДАЛИТИ
3	Адресна доставка	Doe John	Фізика. Основи і механічний рух	Бібліотека ім. Шевченка	03.12.2021 13:00:23	Так	03.01.2022	03.12.2021	Ще не повернуто	ОНОВИТИ ВИДАЛИТИ
16	Адресна доставка	Ковачук Аркадій Андрійович	Полюс Галицький	Бібліотека ім. Лесі Українки	13.12.2021 12:29:28	Так	13.01.2022	13.12.2021	17.12.2021	ОНОВИТИ ВИДАЛИТИ

Кількість замовлень на сторінку 10 1-3 з 3 < >

[Про вебсервіс](#) [Інструкція](#) [API](#)

Рис. 3. Сторінка замовлень

4. Опис створення та редагування замовлення

Натиснувши на кнопку «Замовити» у результатах пошуку користувач переходить на сторінку створення замовлення (рис.3).

Замовити книжку

Найкоротша історія часу



Заповніть інформацію про те, як ви хочете отримати книгу:

Спосіб видачі

Бібліотека

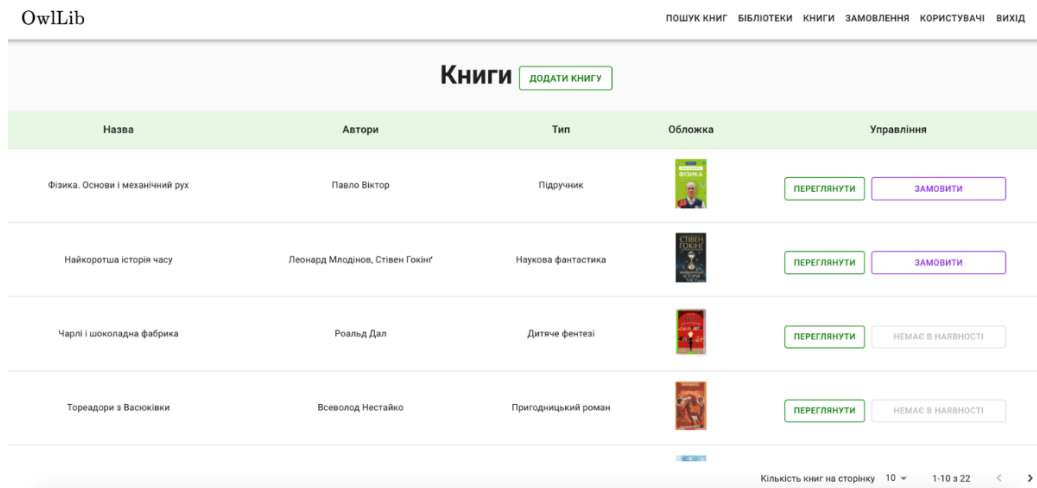
[ЗАМОВИТИ](#)

Рис. 3. Сторінка створення замовлення

Далі користувач обирає спосіб видачі: «у бібліотеці» чи «доставка» та обирає бібліотеку, у якій він би хотів отримати книгу. Замовлення буде перевірене та створене після того, як користувач натисне на кнопку «Замовити» та перейде на сторінку замовлень. Далі адміністратор може натиснути кнопку «Оновити» та змінити замовлення.

5. Опис створення та редагування книг

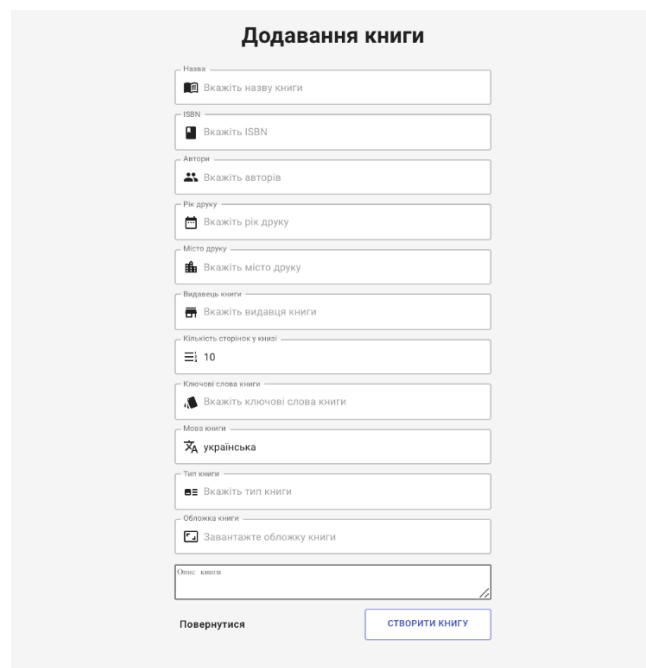
Адміністратор має можливість створити та оновити деталі книги. Для створення книги йому потрібно перейти на сторінку списку книг (рис. 4) та натиснути на кнопку «Додати книгу».



Назва	Автори	Тип	Обложка	Управління
Фізика. Основи і механічний рух	Павло Віктор	Підручник		ПЕРЕГЛЯНУТИ ЗАМОВИТИ
Найкоротша історія часу	Леонард Млоднов, Стівен Гокінг	Наукова фантастика		ПЕРЕГЛЯНУТИ ЗАМОВИТИ
Чарлі і шоколадна фабрика	Розальд Дал	Дитяча фантазія		ПЕРЕГЛЯНУТИ НЕМАЄ В НАВМНОСТІ
Тореадори з Васюківки	Всеволод Нестайко	Пригодницький роман		ПЕРЕГЛЯНУТИ НЕМАЄ В НАВМНОСТІ

Рис. 4. Сторінка списку книг

Після цього його буде перенаправлено на сторінку створення книги (рис. 5).



Додавання книги

Назва

ISBN

Автори

Рік друку

Місто друку

Видавець книги

Кількість сторінок у книзі

Ключові слова книги

Мова книги

Тип книги

Обложка книги

Опис: книги

[Повернутися](#) [СТВОРИТИ КНИГУ](#)

Рис. 5. Сторінка створення книги

Після заповнення форми коректними даними адміністратор зможе створити книгу та перейти на сторінку зі списком книг. При некоректних даних відповідні повідомлення з помилка будуть відображені, а поля пофарбуються у червоний колір.

Для редагування книги адміністратор має перейти на сторінку деталей певної книги, де він зможе натиснути на кнопку «Редагувати». Далі його буде перенаправлено на сторінку редагування книги. Заповнивши форму коректними даними, адміністратор зможе створити книгу та перейти на сторінку зі списком книг. При некоректних даних відповідні повідомлення з помилка будуть відображені, а поля пофарбуються у червоний колір.

ДОДАТОК 5. Приклад презентації



КОМАНДА



Атаманюк Олексій
Розроблення компонент
голосового інтерфейсу



Труш Артем
Графічне моделювання
та анімація



Кравчук Аркадій
Розробка архітектури
і логіки програмної
системи та інтеграція
компонент програмного
продукту

ТЕХНІЧНЕ ЗАВДАННЯ

- Використання голосового вводу для взаємодії з інтерактивним помічником
- Відображення тривимірної анімованої моделі під час використання голосового помічника
- Пошук та фільтрація книг за атрибутами за допомогою голосового вводу
- Замовлення книги для подальшого її отримання у бібліотеці або поштою для авторизованих користувачів

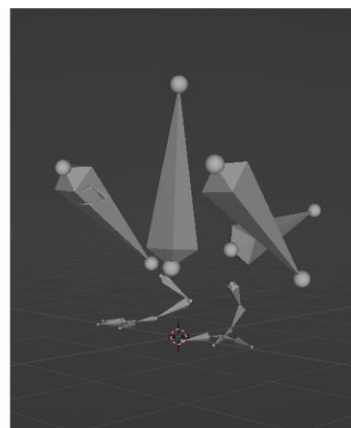
ВИКОРИСТАНІ ТЕХНОЛОГІЇ



3D МОДЕЛЬ



Анімація моделі



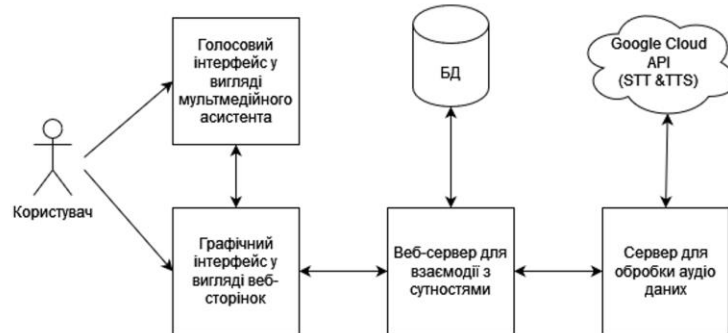
Скелет моделі

ОБРОБКА ГОЛОСУ

Web Audio API



АРХІТЕКТУРА СИСТЕМИ



СТОРІНКА ПОШУКУ КНИГ

OwlLib

ПОШУК КНИГ БІБЛІОТЕКИ КНИГИ ВХІД РЕЄСТРАЦІЯ

Назва

Введіть назву книги

Бібліотеки

Автори

Видавці

Тип

Ключові слова

Мови книг

Роман

5700

5801

Назва	Автори	Тип	Обложка	Управл'
Тореадори з Васюківки	Всеволод Нестайко	Пригодницький роман		ПЕРЕГЛЯНУТИ ЗАМОВИТИ
Різдво приходить у країну Мульти-тролія	Туві Яксон, Сесілія Девідссон, Алекс Гарді	Пригодницький роман		ПЕРЕГЛЯНУТИ ЗАМОВИТИ
Пригоди Тома Соєра	Марк Твен	Пригодницький роман		ПЕРЕГЛЯНУТИ ЗАМОВИТИ
Маленький	Антуан де Сент-	Пригодницький		ПЕРЕГЛЯНУТИ НІМАЄ В НАВІДНОСТІ

Кількість книг на сторінку

10

1-10 з 10

ДЕМОНСТРАЦІЯ



ДЯКУЄМО ЗА УВАГУ!



ПОДЯКА

Укладачі висловлюють подяку випускникам кафедри програмного забезпечення комп'ютерних систем: Олексію Атаманюку, Дмитру Вовченку, Юрію Гришку, Аркадію Кравчуку, Павлу Кривді, Анатолію Санжаровському, Артему Трушу, Петру Тютюнику, Михайлу Чумаку, роботи яких були використані для ілюстрації та прикладів оформлення курсового проєкту.