

Національний технічний університет України
«Київський політехнічний університет імені Ігоря Сікорського»
Інститут прикладного системного аналізу
Кафедра Системного проектування

Лабораторна робота № 4
з предмету «Проектування інформаційних систем»

Виконала: студент
групи ДА-72
Когінова Аліса

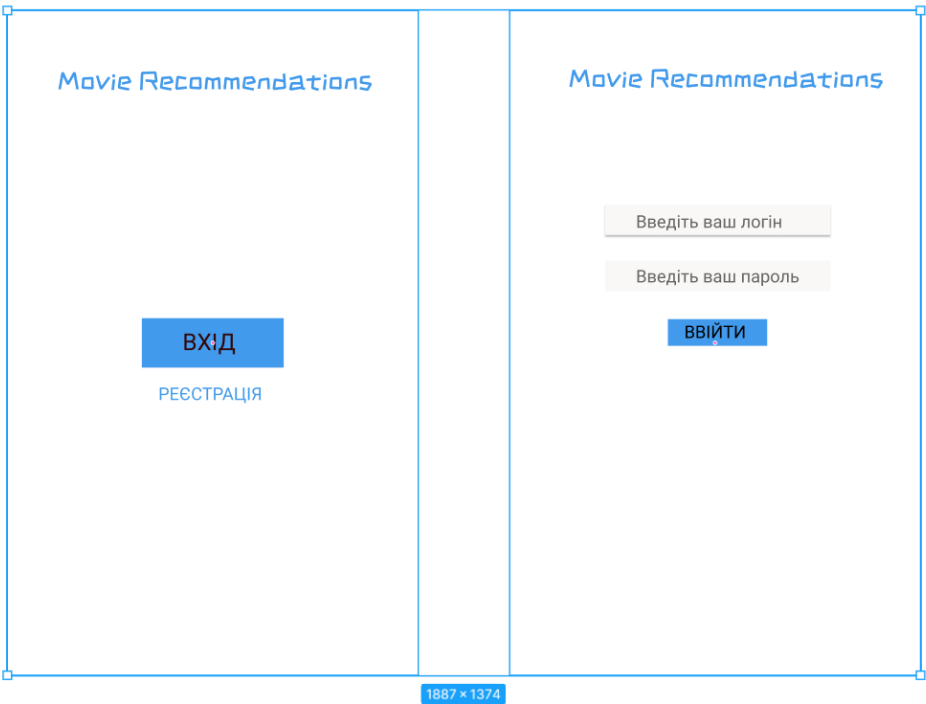
Мета роботи: розробити тестовий мобільний додаток за темою індивідуального завдання.

Задача: Побудувати інтерфейс користувача та функціонал (частково) інформаційної системи за обраною індивідуальною темою. Частково реалізувати функціонал додатку (CRUD).

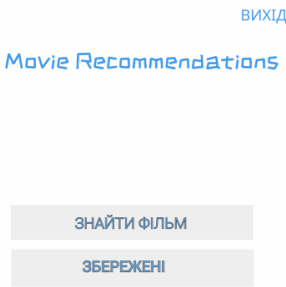
Хід роботи:

1. Інтерфейс користувача додатку

Авторизація/ вхід



Головна сторінка



Знайти фільм та переглянути інформацію про нього

[Вихід](#)

Movie Recommendations


ВВЕДІТЬ НАЗВУ ФІЛЬМА

назва

ЗНАЙТИ

[Вихід](#)

Movie Recommendations



Film information


Directed by
James Cameron


Produced by
James Cameron
Jon Landau


Written by
James Cameron

In the year 2154, Jake Sully (Sam Worthington), a former U.S. Marine paralyzed from the waist down due to wounds sustained in combat, is selected to participate in the Avatar Program. Originally, the position was supposed to be filled by his identical twin brother Tom Sully, who was recently killed by a man who attempted to rob him of his money.

[РЕКОМЕНДАЦІЇ](#)







2. Основний функціонал додатку CRUD:

Архітектура додатку буде складатися з модулів, кожен з яких включатиме у собі сервіси та контролери, з'єднані за допомогою Dependency Injection паттерна у головному модулі.

При викликанні окремого ендпоінту програми у контролері буде йти відповідний виклик сервісного методу, що буде містити бізнес логіку та повертатиме відповідно необхідні дані.

1) CREATE:

Дія: створення нового запису про випиту воду протягом дня

Ендпоінт: головна сторінка додатку, кнопка «Випито!»

Результат дії: до бази даних заноситься новий запис про випиту воду

2) READ:

Дія: зчитування даних для кожного користувача з сервера (бази даних) для синхронізації персональних налаштувань та історії випитої води.

Ендпоінт: головна сторінка додатку, сторінка налаштувань при авторизації користувача; головна сторінка (завантаження історії споживань води) при доданні нового запису (оновлення списку історії).

Результат дії: дані користувача, що зберігаються у базі даних, правильно відображаються на сторінках додатку.

3) UPDATE:

Дія: оновлення даних кожного користувача (налаштувань) на сервері.

Ендпоінт: сторінка налаштувань облікового запису; зміна значень у полях сторінки та збереження за допомогою натискання кнопки «Зберегти».

Результат дії: дані персональних налаштувань користувача оновлені на стороні сервера.

4) DELETE

Дія: видалення запису про випиту воду із історії (на стороні сервера).

Ендпоінт: головна сторінка, кнопка «Видалити» навпроти відповідного запису з «Історії».

Результат дії: відповідний запис видалений з бази даних та не відображається в історії споживань користувача.

Розробити основний функціонал мобільного додатку CRUD

Проведемо тестування ендпоінтів на прикладі ендпоінту /favorites

GET - /favotites – отримуємо список фільмів, які користувач додав у папку favorites

```
{
  {
    "id": 324,
    "name": "Inception"
  },
  {
    "id": 65,
    "name": "Tenet"
  }
}
```

POST - /add_to_favorites/{id} – добавимо фільм у папку favorites

```
{
  "Film 23 was successfully added."
}
```

DELETE - /favorites/{id} – видалення фільму з папки favorites

```
{
  "Film 23 was successfully deleted."
}
```

Випадок, коли фільм з зазначеним id відсутній у папці favorites

```
{  
  "Film 4 was not found in favorites."  
}
```

На усі зазначені запити відповідь серверу була 200 OK, крім останнього запиту, на який сервер відповів: 404 NOT FOUND

3. Висновки:

В ході виконання лабораторної роботи було створено інтерфейс додатку, та досліджено всі нюанси, що можуть виникнути в процесі розробки додатку.

Розроблено план CRUD операцій в додатку – їх основний функціонал та шляхи використання в різних частинах програми.

В середньому час виконання запитів від 70 до 90 мілісекунд, що є досить високим показником як для системи, що з кожним запитом звертається до бази даних SQL, використовуючи SQL.