



Екзаменаційне завдання Створення вебдодатків з використанням Angular і React

Версія 1.0.0

Практична частина іспиту складається з одного завдання. Для його реалізації ви можете використовувати Angular або React. Вибір залишається за студентом.

Перший варіант:

Створити додаток «Список справ». Основне завдання проєкту: надати користувачу можливість керувати своїми справами.

Інтерфейс додатка має надавати такі можливості:

- Додати справу:
 - Назва справи
 - Дата/час виконання
 - Опис справи
 - Теги
 - Пріоритет справи
- Видалити справу
- Редагувати справу
- Збереження списку справ
- Створити проєкт
- Внести справу в проєкт
- Сформувати список справ:
 - На день
 - На тиждень
 - На місяць
- Пошук справ:
 - За назвою
 - За словами в описі
 - За тегом/тегами
 - За пріоритетом

Цей проєкт повинен обов'язково мати клієнтську частину, реалізовану на Angular або React, а також серверну частину, реалізовану на будь-якій технології. Серверна частина необхідна для збереження списку справ та проєктів.

Другий варіант:

Створити додаток «Вікторина». Основне завдання проєкту: надати користувачу можливість проходити вікторини.

Інтерфейс додатка має надавати такі можливості:

- При старті програми користувач вводить логін і пароль для входу. Якщо користувач не зареєстрований, він має пройти процес реєстрації.
- При реєстрації потрібно вказати:
 - Логін (не можна зареєструвати вже існуючий логін);
 - Пароль;
 - Дату народження.
- Після входу в систему користувач може:
 - Стартувати нову вікторину;
 - Переглянути результати своїх минулих вікторин;
 - Переглянути топ-10 за конкретною вікториною;
 - Змінити налаштування. Можна змінювати пароль та дату народження;
 - Вихід.
- Для старту нової вікторини користувач повинен обрати розділ знань вікторини. Наприклад, "Історія", "Географія", "Біологія" і т. д. Також потрібно передбачити змішану вікторину, коли питання будуть обиратись із різних вікторин за випадковим принципом.
- Конкретна вікторина складається із двадцяти питань. Кожне питання може мати один або декілька правильних варіантів відповідей. Якщо питання передбачає декілька правильних відповідей, а користувач вказав не все, питання не зараховується.
- Після завершення вікторини користувач отримує кількість правильних відповідей, а також отримує своє місце у таблиці результатів гравців вікторини.

Цей проєкт повинен обов'язково мати клієнтську частину, реалізовану на Angular або React, а також серверну частину, реалізовану на будь-якій технології.

Третій варіант:

Створіть гру «Пам'ять». Ознайомитися з правилами гри англійською можна за цим посиланням:

<https://www.memozor.com/memory-games/memory-game-rules>

Головні особливості реалізації гри:

- Вибір розміру поля
- Початкове розміщення карток виконується комп'ютером
- У кожного користувача гри є свій логін і пароль для входу
- Звуковий супровід гри:
 - Якщо картки збігаються, звучить один звук
 - Якщо картки не збігаються, звучить інший звук
- Статистика:
 - Час гри
 - Кількість переворотів карток
- Вибір стильового оформлення гри:
 - Темна тема
 - Світла тема
- Таблиця рекордів

Цей проєкт повинен обов'язково мати клієнтську частину, реалізовану на Angular або React, а також серверну частину, реалізовану на будь-якій технології.

Четвертий варіант:

Створіть гру «Морський бій». Ознайомитися з правилами гри можна за цим посиланням: [https://uk.wikipedia.org/wiki/Морський_бій_\(настільна_гра\)](https://uk.wikipedia.org/wiki/Морський_бій_(настільна_гра))

Головні особливості реалізації гри:

- Режими гри:
 - Людина — комп'ютер
 - Комп'ютер — комп'ютер
 - Людина — людина
- Розміщення кораблів:
 - Гравець розставляє вручну
 - Комп'ютер розставляє замість гравця
- Комп'ютер може грати у двох режимах:
 - Випадковий постріл
 - Інтелектуальна гра. Комп'ютер будує стратегію гри, а не стріляє випадково
- Відображати поточний стан ігрового поля гравця та комп'ютера
- Можливість зупинки, припинення, нового запуску гри
- Вибір стильового оформлення гри:
 - Темна тема
 - Світла тема
- У кожного користувача гри є свій логін і пароль для входу
- Таблиця рекордів

Цей проєкт повинен обов'язково мати клієнтську частину, реалізовану на Angular або React, а також серверну частину, реалізовану на будь-якій технології.

П'ятий варіант:

Створити додаток «Бібліотека». Основне завдання проєкту: надати користувачу можливість знаходити та читати безкоштовні книги.

Інтерфейс додатка має надавати такі можливості:

- Пошук книги:
 - За автором
 - За назвою
 - За жанром
- Пошук усіх книг автора
- Пошук усіх книг певного жанру
- Читати обрану книгу
- Завантажити обрану книгу
- Додати книгу. При додаванні книги потрібно додати наступну інформацію:
 - Назва книги
 - Автор

- Жанр
 - Рік видання
 - Текст книги (у вигляді текстового файлу)
- Редагувати інформацію про книгу
- Оцінити книгу
- Написати відгук по книзі
- Вивести рейтинг книг:
 - 10 найпопулярніших книг у всіх жанрах
 - 10 найпопулярніших книг у певному жанрі
 - 3 найпопулярніші книги певного автора

Цей проєкт повинен обов'язково мати клієнтську частину, реалізовану на Angular або React, а також серверну частину, реалізовану на будь-якій технології.