**Дисципліна «Основи програмної інженерії»**

**Лабораторна робота№10-11**

**Виконав студент**

**Групи 23-ІПЗ**

**Бреус Артем**

# Завдання№1 (Короткий опис продукту)

*Ми налаштовані* на створення сайту з метою покращити досвід користувачів під час пошуку фільмів з подальшим придбанням квитка.

Плануємо створити нові привілеї для зареєстрованих користувачів, щоб більше людей реєструвалось та поверталось до нас.

Як ми вже звикли, кожного дня може відбутися повітряна тривога. Тому із-за цього перегляд кіно припиняється та життя переглядачів може бути у небезпеці.

Я вважаю, ми маємо декілька спосіб вирішити проблему, якщо переглядач має зареєстрований облік, то він має змогу – 1 варіант Отримати доступ до перегляду у онлайн форматі 1 раз (Завантаження звісно ж заборонене), 2 варіант Дати можливість прийти на інший день та подивитись кіно (але це вже існує на даний час) І останній 3 варіант – Створити спеціальне приміщення для перегляду фільмів, навіть під час повітряної тривоги і бути в безпеці.

# Завдання№2 (Структура ТЗ)

## **Найменування та Область застосування**:

Найменування:

Наш програмний продукт має назву «CinemaVerse», продукт спрямований на перегляд фільмів у затишку та безпеці.

Область Застосування:

Використовується через мережу Інтернет, але спрямований на перегляд перед великим екраном, його використовують як працівники кінотеатру (за касою кінотеатру під час оформлення квитка), так і користувачі.

## Підстава для розробки:

**Підставою для розробки** є метою завдання з дисципліни «Основи програмної інженерії», але для нас це завдання спрямовує на розвинення власних навичок у створенні проектів та доцільних звітів

Документ створено відповідно до:

1. ISO/IEC 29148:2018 — Systems and software engineering — Life cycle

processes — Requirements engineering

2. SO/IEC 25010:2011 — Systems and software quality requirements and

evaluation

3. Agile Manifesto

4. Scrum Guide 2020

## Призначення розробки:

Функціональне призначення:

Система забезпечує такий функціонал:

* Створення облікового запису
* Перегляд розкладу фільмів
* Придбання квитків Онлайн
* Перегляд контакту менеджера

Експлуатаційне призначення:

Додаток має використовуватись через браузер (Google Chrome, Mozzila, Edge…)

Підтримує використання сторінки з комп’ютера, телефона, планшету…

## Технічні вимоги:

Технологічний стек:

* Frontend: React, react-router, Vite, TailwindCSS, CSS.
* Backend: Node.js
* База даних: SQLite
* Репозиторій: GitHub

Модулі:

* Модуль Автентифікації
* Модуль каталогу фільмів
* Модуль розкладу сеансів
* Модуль бронювання місць з оплатою
* Модуль квитків

Функціональні вимоги:

|  |  |  |
| --- | --- | --- |
| № | Функція | Опис |
| Fn1 | Огляд афіші | Список з фільмами |
| Fn2 | Пошук-Фільтрація | Пошук за назвою та фільтрувати за датою, жанром, тривалістю фільму |
| Fn3 | Перегляд сеансів | Перегляд доступних до перегляду фільмів |
| Fn4 | Бронювання місць | обрання фільму та місця |
| Fn5 | Оплата квитків | Оплата карткою через мережу Інтернет |
| Fn6 | Створення квитка | Генерація квитку у форматі PDF з QR-кодом |
| Fn7 | Особистий кабінет | Історія переглянутих фільмів, бонуси |

Умови експлуатації:

Система пристрою: будь-яка, має підтримувати браузер. Тобто Windows 10+, MacOS, Linux

Браузер: Chrome, Firefox, Edge, Opera GX…

Підключення до Інтернету має бути не менше 3 Mbps+

Вимоги до технічних засобів:

|  |  |  |
| --- | --- | --- |
| Компонент | Мінімум | Рекомендовано |
| Процесор | 2 cores, 2 threads, 2 GHz | 4 cores, 8 threads, 2.5 GHZ |
| Оперативна пам’ять, RAM | 2 GB | 4 GB |
| HDD/SSD | 1,7 GB Для користувача, 20 GB для розробника | 3 GB Для користувача, 30+ GB для розробника |

## Стадії розробки:

|  |  |  |
| --- | --- | --- |
| Спринт, № | Тривалість | Поставлена мета |
| 1 | 2 тижні | Аналіз вимог, макетування сторінок (Fn1, Fn2) |
| 2 | 2 тижні | Рекреація основних макетів та функцій, підключення бази даних (Fn3, Fn4) |
| 3 | 2 тижні | Інтеграція оплати та створення квитка (подальша можливість друкувати), створення особистого кабінету користувача (Fn5, Fn6, Fn7) |
| 4 | 2 тижні | Тестування, дороблення нереалізованих функцій і виправлення помилок в коді, документування виконаної роботи та відкриття сайту |

## Порядок тестування та приймання:

Види тестування:

Функціональне тестування – покупка/бронювання.

Unit-тести (Jest).

Integration-тести (Postman).

UI-тест – (Playwright).

Користувацьке тестування – Тестування з різних пристроїв, та прийнятної кількості користувачів водночас.

ПЗ вважається прийнятним якщо:

Всі важливі функції (купівля, бронювання)працюють коректно

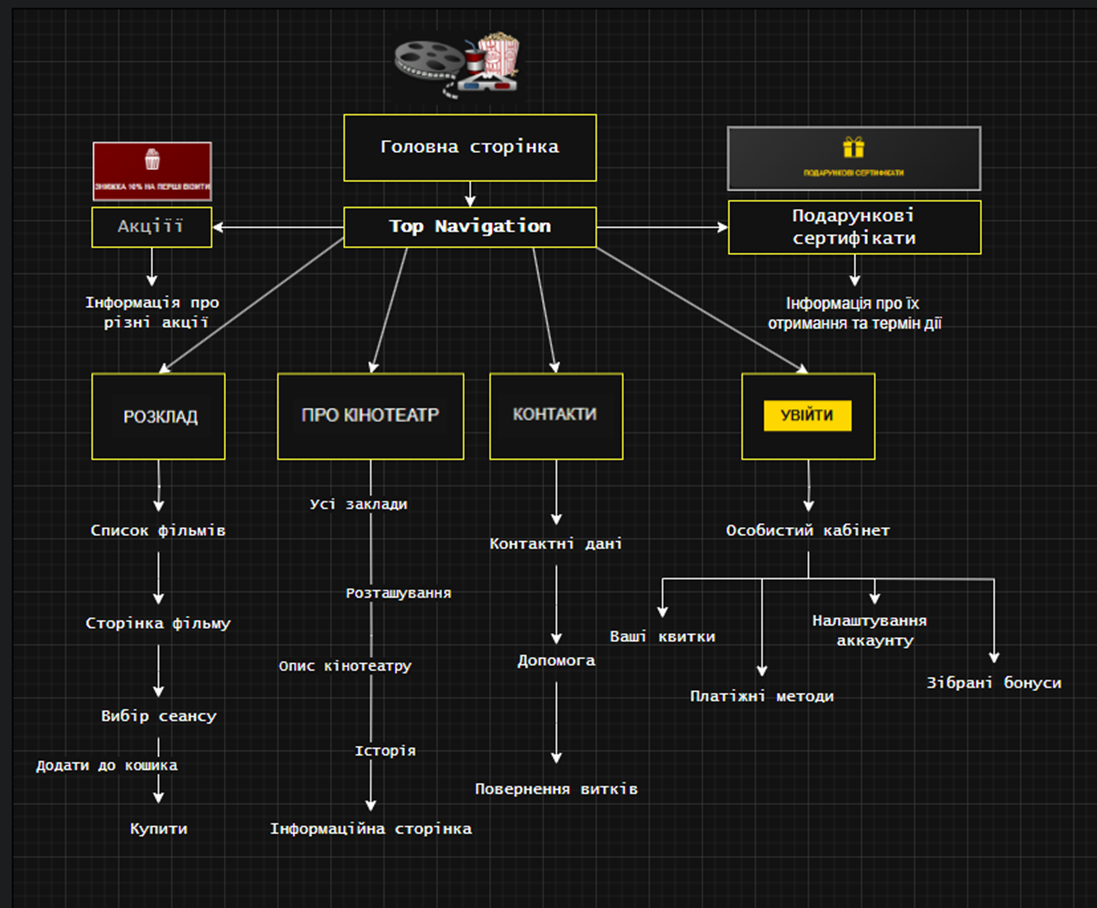
Дані про купівлю мають зберігатися в базі даних, створює квиток

Інтерфейс відображається коректно на всіх пристроях

Відсутність критичних багів

# Завдання№3 (Додатки)

“Логічна структура сайту”



“Діаграма «Use Case»”

