ПРОЕКТ  
В КАТЕГОРИЯ

Софтуерни Приложения

ТЕМА: Бизнес и/или социален проблем

Автор:

Ивайло Славчев Бъчваров 10 Б

Иван Янков Стойчев 10 Б

Ръководител:

Димитър Янакиев

БургасСЪДЪРЖАНИЕ

[1 РЕЗЮМЕ 2](#_Toc263042848)

[2 Цели (предназначение, кратък анализ на потребностите и на съществуващите решения) 3](#_Toc1148157107)

[3 Основни етапи в реализирането на проекта (основни дейности, роли на авторите) 4](#_Toc1803415783)

[Потребителски изисквания и работен процес 4](#_Toc913868975)

[Примерен потребителски интерфейс 4](#_Toc1315308822)

[Модел на съдържанието / данните 4](#_Toc1598254760)

[4 Ниво на сложност на проекта − основни проблеми при реализация на поставените цели 5](#_Toc1660749643)

[5 Логическо и функционално описание на решението – архитектура, от какви модули е изградено, какви са функциите на всеки модул, какви са взаимодействията помежду им и т.н. 5](#_Toc95058841)

[6 Реализация − обосновка за използвани технологични средства, алгоритми, литература, програмни приложения и др. 6](#_Toc1530806235)

[7 Описание на приложението – как се стартира и/или инсталира, как се използва, как се поддържа 8](#_Toc1258534926)

[8 Заключение и възможно бъдещо развитие 10](#_Toc654115385)

[9 Използвани литературни източници и Уеб сайтове 10](#_Toc1824729147)

[10 Приложения 10](#_Toc693087481)

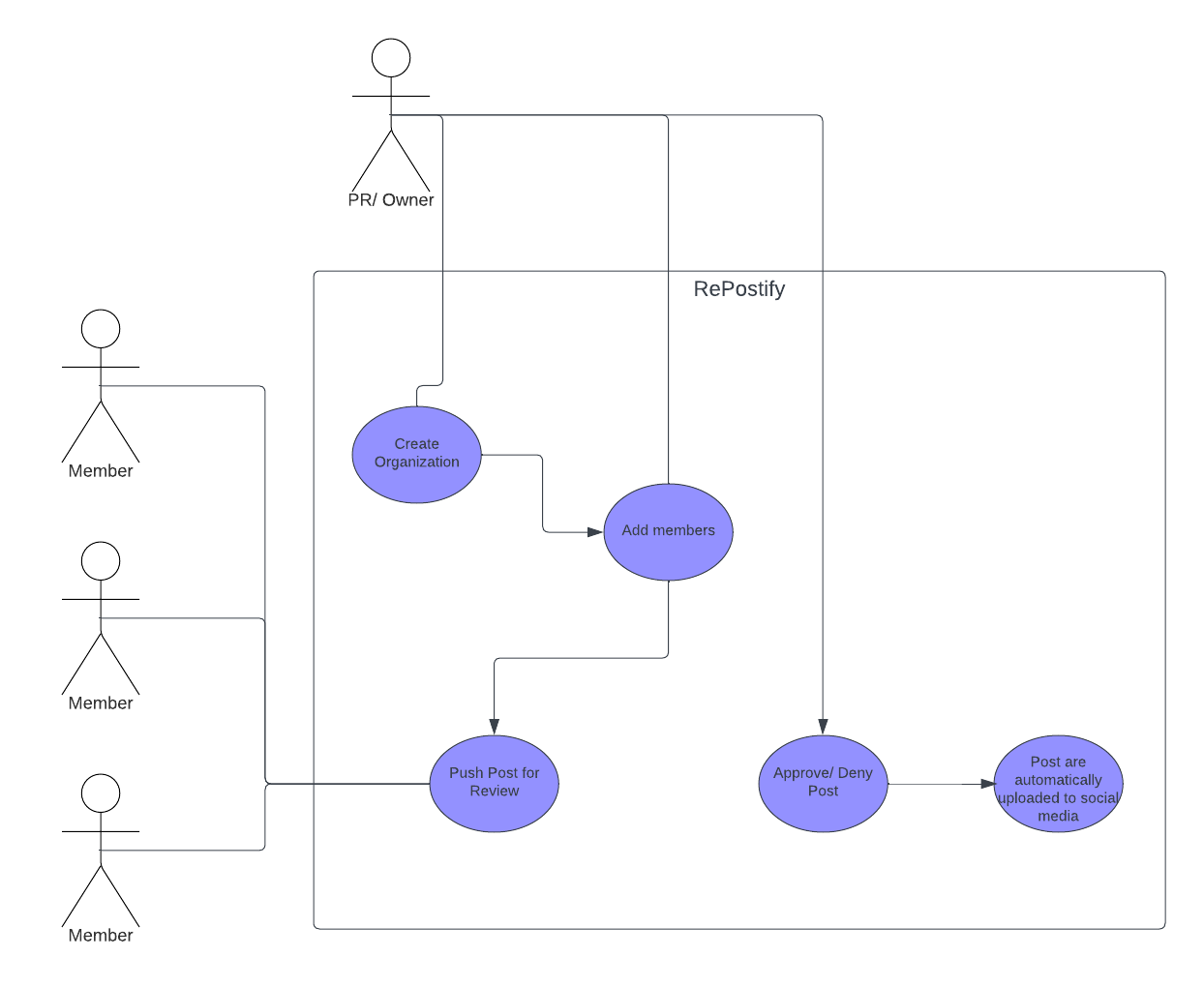
[11 КРИТЕРИИ ЗА ОЦЕНЯВАНЕ 12](#_Toc1335549775)

[СОФТУЕРНИ ПРИЛОЖЕНИЯ 12](#_Toc1990468365)

[РАЗПРЕДЕЛЕНИ ПРИЛОЖЕНИЯ 13](#_Toc629018181)

# РЕЗЮМЕ

Избрахме да направим web приложение на тема бизнес проблем, което опростява менажирането на постове в социалните мрежи на компании, за да улесним работата на PR мениджърите.



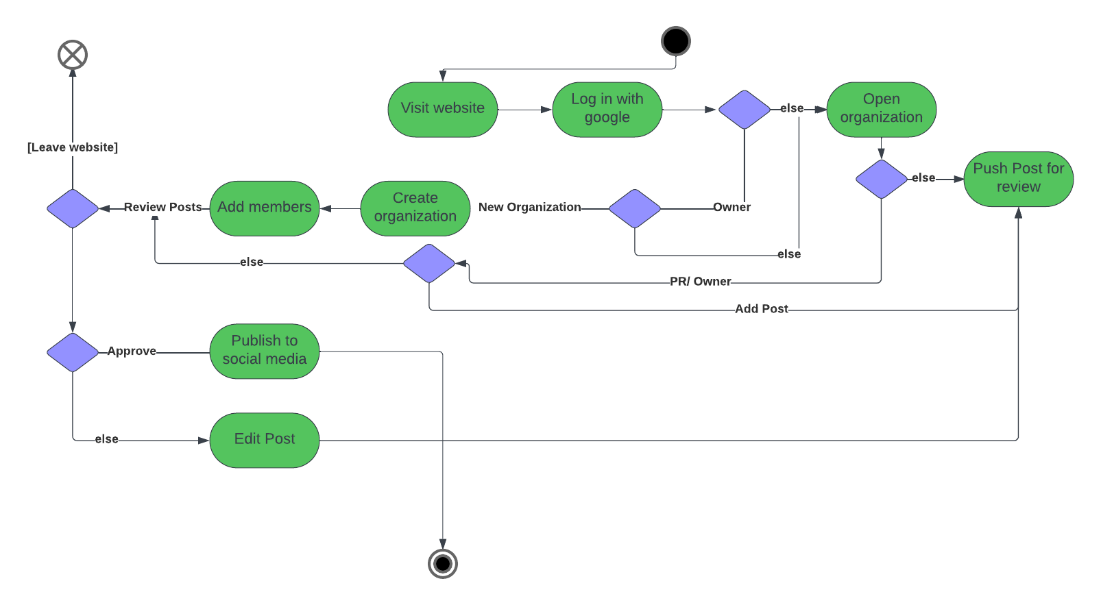
# Цели (предназначение, кратък анализ на потребностите и на съществуващите решения)

Проектът представлява иновативно решение за ефективно управление на социалните мрежи чрез централизирана платформа. Потребителите имат възможността да създават и персонализират социални постове, определяйки тяхната целева аудитория и контекст. След подготовка, постовете се предоставят за одобрение на отговорни лица като PR или собствениците на организацията.

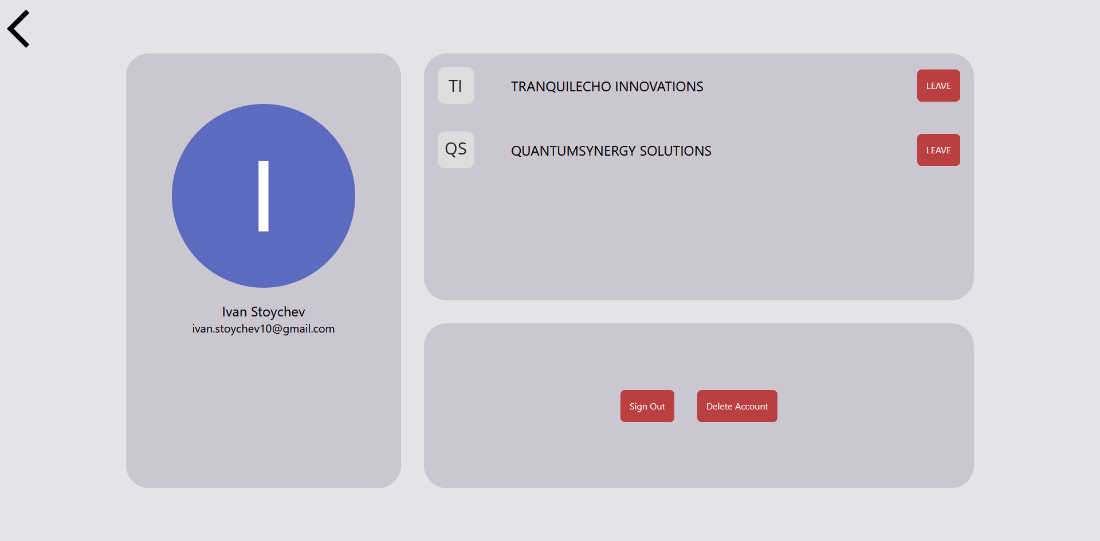
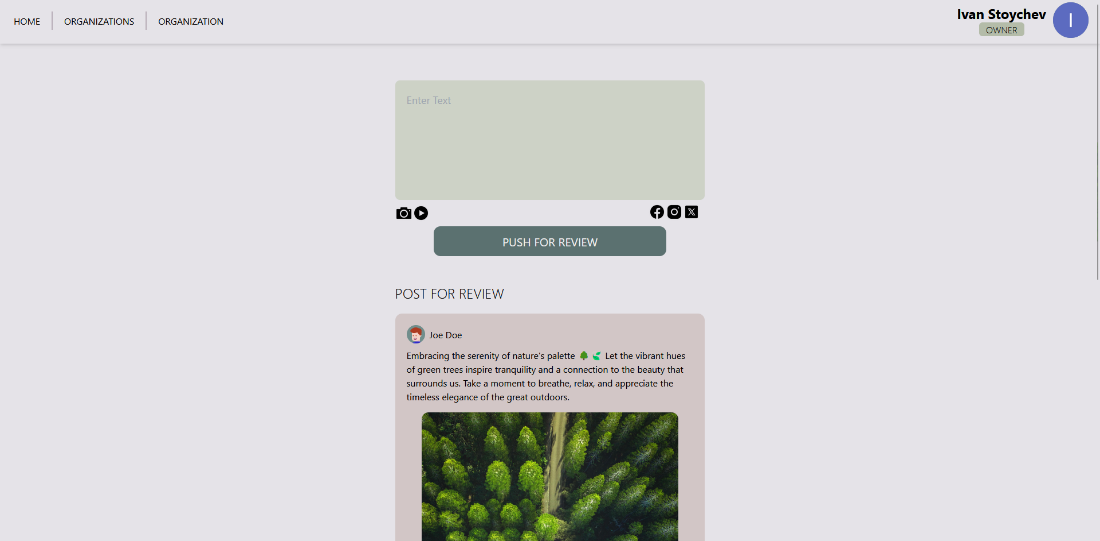
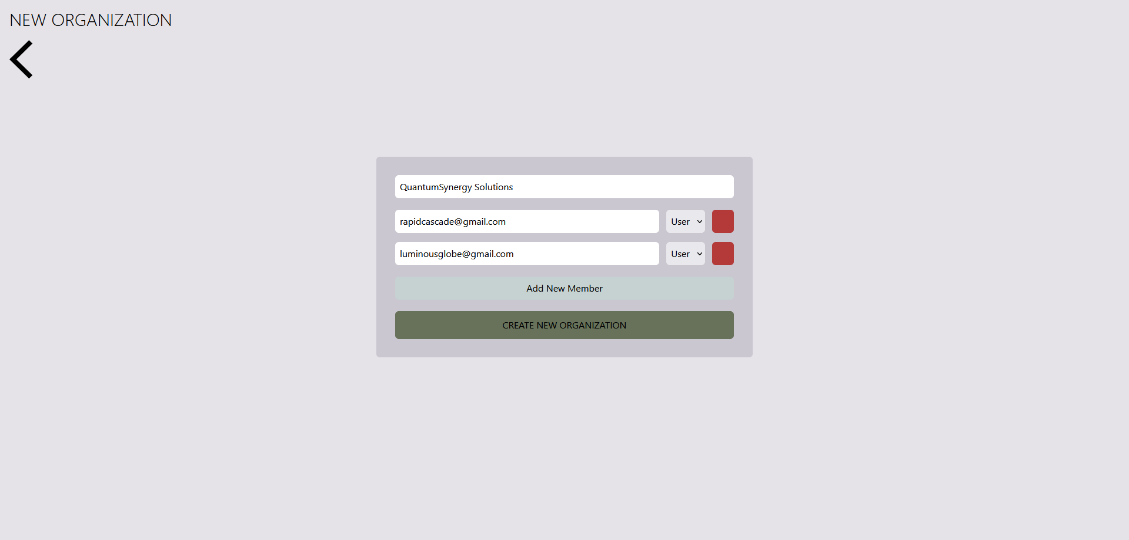
Одобрените постове автоматично се разпределят и публикуват в съответните социални мрежи, улеснявайки процеса на мениджмънт на контента. Този подход гарантира не само ефективност в управлението на социалните мрежи, но и повишена сигурност чрез предварително одобрение на съдържанието. В резултат на това, организациите се възползват от по-структурирано и целенасочено онлайн присъствие, допринасяйки за подобрена репутация и обществена видимост.

# Основни етапи в реализирането на проекта (основни дейности, роли на авторите)

## Потребителски изисквания и работен процес



## Примерен потребителски интерфейс



## Модел на съдържанието / данните

* Потребителски профил:
  + Име
  + Потребителско име
  + Имейл адрес
  + Парола
  + Длъжност/Роля (Потребител, PR, Owner)
* Съдържание на поста:
  + Текст на поста
  + Вложения (картини, видеа, документи)
  + Хаштагове
  + Линк към статия или уебсайт
* Целева аудитория:
  + Демографски данни (възраст, пол, местоположение и др.)
  + Интереси
  + Потребителски групи
* Избор на социални мрежи:
  + Facebook
  + Twitter
  + Instagram
  + LinkedIn
  + Други
* Одобрение на поста:
  + Състояние на одобрение (чака одобрение, одобрен, отхвърлен)
  + Одобрено от (PR, Owner)
  + Дата и час на одобрение
* История на постовете:
  + Списък с всички създадени и публикувани постове
  + Състояние на публикацията (публикуван, не е публикуван)
  + Дата и час на публикуване

# Ниво на сложност на проекта − основни проблеми при реализация на поставените цели

// TODO: сорс код

Тази секция представя дизайна на решението на проблема за проекта ви. Опишете каква софтуерна платформа сте избрали за вашето решение /напр. .NET, java/. Представете схема на софтуерната архитектура на решението /по модули и/или слоеве/ с диаграма на разгръщането, както и диаграми на класовете на дизайна /с ограничения, описани на OCL/, диаграми на времето /за задаване на времена за синхронизация и комуникация в решението/ и компонентни диаграми. Илюстрирайте решението с извадки от генериран сорс код.

# Логическо и функционално описание на решението – архитектура, от какви модули е изградено, какви са функциите на всеки модул, какви са взаимодействията помежду им и т.н.

* **Фронтенд (Frontend):**
  + **Модул "Потребителски интерфейс":**
    - Функции: Показва потребителския интерфейс, където потребителите могат да създават, редактират и изтриват задачи.
    - Взаимодействия: Комуникира със задачите на потребителя и изпраща заявки към бакенда за обновяване на данните.
* **Бакенд(Backend):**
  + **Модул "Управление на задачи":**
    - Функции: Обработва логиката на приложението, включително създаване, редактиране, изтриване и списък с задачи.
    - Взаимодействия: Комуникира с базата данни за съхранение на информацията за задачите.
  + **Модул "Автентикация и авторизация":**
    - Функции: Управлява потребителските идентификации и правата за достъп.
    - Взаимодействия: Проверява потребителските данни и управлява сесиите.
  + **Модул "База данни":**
    - Функции: Съхранява данни за потребителите и задачите.
    - Взаимодействия: Получава и записва информацията, предоставена от другите модули.
* **Инфраструктура:**
  + **Модул "Сървър":**
    - Функции: Обработва HTTP заявки и осигурява връзка между фронтенда и бакенда.
    - Взаимодействия: Приема HTTP заявки от фронтенда и ги препраща към съответния модул на бакенда.
  + **Модул "База данни сървър":**
    - Функции: Управлява връзката с базата данни и извършва операции за четене и запис на данни.
    - Взаимодействия: Получава заявки от модула за управление на задачи и модула за автентикация и авторизация.

# Реализация − обосновка за използвани технологични средства, алгоритми, литература, програмни приложения и др.

За реализацията на конкретен софтуерен проект са необходими различни технологии и инструменти. Ние избрахме долуизброените тегнологични средства спрямо нашите потребности и длъжности.

* **Технологични средства:**
  + **Език за програмиране (Backend/Frontend):**
    - Backend
      * Python (FastAPI)
    - Frontend
      * JavaScript (React)
  + **База данни:** 
    - PostgreSQL.
  + **Сървър:**
    - Azure
* **Инструменти за управление на зависимости и сборка:**
  + npm
  + Pip
* **Литература:**
  + **Документация на използваните технологии:** Официалната документация на езиците за програмиране, библиотеките и фреймуърките.
  + **Книги и онлайн ресурси:**
    - [FastAPI](https://fastapi.tiangolo.com/)
    - [React](https://react.dev/)
* **Програмни приложения:**
  + **Интегрирани среди за разработка (IDE):**
    - Visual Studio Code
    - Docker
  + **Версионен контрол:**
    - Git
    - Github
  + **Базови програми за дизайн и разработка:**
    - Figma
    - Postman
    - PgAdmin 4

# Описание на приложението – как се стартира и/или инсталира, как се използва, как се поддържа

* **Инсталация и стартиране:**
  + **Клониране на репозиторито:**
    - **git clone** [**https://github.com/IYStoychev21/RePostify**](https://github.com/IYStoychev21/RePostify)
  + **Инсталиране на зависимостите:** 
    - **install–Windows.bat** за Windows системи
    - **install-Linux.sh** за Linux системи
  + **Стартиране на бакенда, бакенда и базата данни:**
    - **docker compose up**
* **Използване на приложението:**
  + **Регистрация и вход:**
    - Потребителите трябва да се регистрират с валиден имейл и парола.
    - Вече регистрирани потребители влизат в системата чрез въвеждане на потребителско име и парола.
  + **Управление на задачи:**
    - Създаване на нова задача с указване на заглавие, описание, приоритет и дата на изпълнение.
    - Редактиране и изтриване на съществуващи задачи.
    - Филтриране и търсене на задачи по различни критерии.
  + **Потребителски интерфейс:**
    - Интуитивен и лесен за използване интерфейс.
    - Достъп до различни секции като "Всички задачи", "Завършени задачи", "Неприключени задачи" и т.н.
    - Визуални индикации за приоритет и статус на задачите.
* **Поддръжка и развитие:**
  + **Документация:**
    - Подробна документация за инсталация, конфигурация и използване на приложението.
    - Обновена и пълна документация за кода, включително коментари и обяснения на сложни алгоритми или решения.
    - Swagger документация за ендпойнтовете на бакенда
  + **Тестване:**
    - Автоматизирани тестове за основната функционалност на приложението.
    - Редовно изпълнение на тестове за увереност в стабилността на системата.
  + **Мониторинг:**
    - Интегриране на система за мониторинг, която следи здравето на приложението и предупреждава за евентуални проблеми.
  + **Обновявания и поправки:**
    - Редовни обновления с нови функционалности и подобрения.
    - Бързо реагиране на потенциални сигнали за проблеми и издаване на поправки.
  + **Поддръжка на сигурност:**
    - Редовно актуализиране на зависимостите и прилагане на препоръчителни практики за сигурност.
    - Разглеждане и отстраняване на възможни уязвимости.

# Заключение и възможно бъдещо развитие

В бъдеще може приложението да се развие чрез добавяне на изкуствен интелект за подобряване и анализ на дейността в социалните мрежи.

# Използвани литературни източници и Уеб сайтове

(React Documentation, n.d.)

(FastAPI Documentation, n.d.)

# КРИТЕРИИ ЗА ОЦЕНЯВАНЕ

## СОФТУЕРНИ ПРИЛОЖЕНИЯ

|  |  |  |
| --- | --- | --- |
| **Критерии** | **Точки** | **Макс. т.** |
| **Общи характеристики:**   * Оригиналност на избраната тема и/или на предложеното решение * Яснота, комплексност и значимост на поставените цели * Пълнота на предоставената функционалност * Степен на завършеност, достоверност, актуалност * Приложимост * Използват се нови технологии или по иновативен начин |  | 20 |
| **Проектиране:**  Подходящо избрани технологии   * използват се по предназначение * покриват изискванията на задачата * използват се ефективно Подходящо избрана архитектура * добре структурирани слоеве и програмни модули * разделяне на визуализацията от бизнес логиката * осигурява стабилност и сигурност * използва се ефективно   **Потребителски интерфейс и изживяване** (UI/UX)   * ясна навигация или откриване на нужната функционалност * естетическо оформление * взаимодействие с потребителя |  | 25 |
| **Реализация:**  Качество на изпълнението:   * Именуване на програмни единици (променливи, функции и др.) * Подреденост и четливост на кода и наличие на коментари * Спазване на стандарти, защита на данни, валиден код * Ефективно използване на ресурсите Качество на системата: * Бързо действие – оптимизация на кода на ниво изпълнение * Лесна инсталация и експлоатация * Ниво на стабилност и сигурност, защита от популярни атаки * Адекватна обработка на грешки и непредвидени събития   **Потребителски интерфейс и изживяване** (UI/UX)   * Разпределение на компонентите * Авторски компоненти и дизайн * Лесно четене, въвеждане и възприемане на информацията * Спазени са книжовно-езиковите норми * Съвместимост, разделителна способност на екрана, цветови режими |  | 35 |
| **Представяне и защита на проекта**   * валидиране и верифициране (доказателства от тестване) * наличие на потребителска документация * добре структурирано описание на системата (документацията за НОИТ) * рекламни материали * представяне (поведение, спазване на време, акцент върху важните неща) * защита (отговаряне на въпроси, представяне на аргументи) |  | 20 |
| Общо: |  | 100 |

## РАЗПРЕДЕЛЕНИ ПРИЛОЖЕНИЯ

|  |  |  |
| --- | --- | --- |
| **Критерии** | **Точки** | **Макс. т.** |
| **Общи характеристики:**   * Оригиналност на избраната тема и/или на предложеното решение * Яснота, комплексност и значимост на поставените цели * Пълнота на предоставената функционалност * Степен на завършеност, достоверност, актуалност * Приложимост * Използват се нови технологии или по иновативен начин |  | 20 |
| **Проектиране:**  Подходящо избрани технологии   * използват се по предназначение * покриват изискванията на задачата * използват се ефективно Подходящо избрана архитектура * добре структурирани слоеве и програмни модули * разделяне на визуализацията от бизнес логиката * осигурява стабилност и сигурност * използва се ефективно   **Потребителски интерфейс и изживяване** (UI/UX)   * ясна навигация или откриване на нужната функционалност * естетическо оформление * взаимодействие с потребителя |  | 25 |
| **Реализация:**  Качество на изпълнението:   * Именуване на програмни единици (променливи, функции и др.) * Подреденост и четливост на кода и наличие на коментари * Спазване на стандарти, защита на данни, валиден код * Ефективно използване на ресурсите Качество на системата: * Бързо действие – оптимизация на кода на ниво изпълнение * Лесна инсталация и експлоатация * Ниво на стабилност и сигурност, защита от популярни атаки * Адекватна обработка на грешки и непредвидени събития * Използване на система за контрол на версиите * Автоматично тестване и проверка   **Потребителски интерфейс и изживяване** (UI/UX)   * Разпределение на компонентите * Авторски компоненти и дизайн * Лесно четене, въвеждане и възприемане на информацията * Спазени са книжовно-езиковите норми * съвместимост, разделителна способност на екрана, цветови режими |  | 35 |
| **Представяне и защита на проекта**   * валидиране и верифициране (доказателства от тестване) * наличие на потребителска документация * добре структурирано описание на системата (документацията за НОИТ) * рекламни материали * представяне (поведение, спазване на време, акцент върху важните неща) * защита (отговаряне на въпроси, представяне на аргументи) |  | 20 |
| Общо: |  | 100 |