ПРОЕКТ В КАТЕГОРИЯ

СОФТУЕРНИ ПРИЛОЖЕНИЯ

ТЕМА: САМОУЧИТЕЛ

Автор:

Иван Светославо Делчилов

Димитър Калоянов Бяндов

Ръководител:

Димитър Янакиев

Бургас

# 1 РЕЗЮМЕ

SHS е уеб базирана платформа. Проектът е разработен за и в полза на ученици, които имат необходимостта да получат повече информация по дадена тема и/или да разрешат проблем свързан с неразбиране на учебен материал. Платформата представлява своеобразно приложение, в което ученикът търсещ информация задава конкретен въпрос или проблем до регистрираните в платформата потребители - ученици. Темите /въпросите/ могат да бъдат свързани само с учебния материал изучаван в средните училища/от 8 до 12 клас/. Поради това, че обменът на информация се осъществява единствено и само между ученици, името на платформата е SHS (Students Help Students).

В изложените по-долу точки от настоящият документ са описани по-подробно целите на предложения проект, етапите от разработване на всеки един елемент на платформата и тяхното ниво на сложност, логическото и функционалното описание на решението, както и неговата структура, приложение и възможност за надграждане.

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

Основна цел на проекта е да се създаде платформа в помощ на ученици в гимназиален етап на обучение.

Подцелите на решението са:

* Да се подобрят знанията и уменията на учениците по дадени теми/ въпроси/;
* Да се осъществи една лесно достъпна и разбираема по отношение на комуникацията добра практика - „от връстници към връстници“;
* Ученици имащи интереси към учебни дисциплини, които не изучават, да имат възможност да се поучат от знанията и опита от тези, които ги изучават.

Платформата е интернет приложение. Приложима е за мобилни устройства – смарт телефони. Има форма за регистрация, както и такава за анулиране на акаунта.

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

Първият етап от процеса на създаване на платформата е формиране на структурата й и определяне на функционалното предназначение на всеки един елемент от нея

## Потребителски изисквани и работен процес

## Примерен потребителски интерфейс

## Модел на съдържанието / данните

# 4 Ниво на сложност на проекта - основни проблеми при реализация на поставените цели

# 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.

* **Фронтенд (Frontend):**
  + **Модул "Потребителски интерфейс":**
    - Функции: Показва потребителския интерфейс, където потребителите могат да създават, коментират по дадените запитвания
    - Взаимодействия: Комуникира със постовете на потребителя и изпраща заявки към бекенда за обновяване на данните.
* **Бекенд (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 постовете.
  + **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 постовете.
    - Взаимодействия: Получава и записва информацията, предоставена от другите модули.
* **Инфраструктура:**
  + **Модул "Сървър":**
    - Функции: Обработва HTTP заявки и осигурява връзка между фронтенда и бекенда.
    - Взаимодействия: Приема HTTP заявки от фронтенда и ги препраща към съответния модул на бекенда.
  + **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.

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

* **Технологични средства:**
  + **Език за програмиране (Backend/Frontend):** Програмният език който сме избрали za Backend е TypeScript (TS). За създаването на Frontend използвахме JavaScript библиотеката React. При създаването на нашия Backend . В направата на нашия проект бяха прибавени – Axios, за свързването на Frontend-а и Prisma, за Свързването на Backend-a с базата данни.
  + **База данни:** Релационната база данни която сме избрали е PostgreSQL.
  + **Сървър:** За сървърната част използвахме уеб сървъра ExpressJS (за TypeScript).
  + **Инструменти за управление на зависимости и сборка:** За управлението на зависимости и сборка използваме npm
* **Алгоритми:**
  + **Алгоритми за управление на задачи:** Например, алгоритми за търсене, сортиране и филтриране на задачи, както и алгоритми за валидация на данни.
  + **Алгоритми за сигурност:** Включване на алгоритми за криптиране и управление на достъпа за защита на потребителските данни.
* **Литература:**
  + **Документация на използваните технологии:** Официалната документация на езиците за програмиране, библиотеките и фреймуърките.
  + **Книги и онлайн ресурси:** За допълнително разбиране на концепциите и практиките в областта на уеб разработката и софтуерната архитектура.
* **Програмни приложения:**
  + **Интегрирани среди за разработка (IDE):** Средата за разработка на проекта SHS беше Visual Studio Code.
  + **Версионен контрол:** За контролирането на кода сме използвали Git и GitHub
  + **Базови програми за дизайн и разработка:** Базовите програми за дизайн са Figma, Adobe и Photoshop.

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

* **Инсталация и стартиране:**
  + **Стартиране на бекенда:**
    - Клониране на репозиторито: **git clone** [**https://github.com/your/backend-repo.git**](https://github.com/your/backend-repo.git)
    - Инсталиране на зависимостите: **npm install** или **pip install -r requirements.txt**
    - Запуск на сървъра: **node server.js** или **python manage.py runserver**
  + **Стартиране на фронтенда:**
    - Клониране на репозиторито: **git clone** [**https://github.com/your/frontend-repo.git**](https://github.com/your/frontend-repo.git)
    - Инсталиране на зависимостите: **npm install**
    - Стартиране на разработъчния сървър: **npm start**
  + **База данни:**
    - Конфигуриране и стартиране на база данни (PostgreSQL, MySQL, MongoDB и т.н.)
* **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 нов пост с указване на даден казус по даден учебен предмет, името на предмета, описание на затруднението, дата на качване на поста .
    - Редактиране и изтриване на съществуващи постове.
  + **Потребителски интерфейс:**
    - Интуитивен и лесен за използване интерфейс.
    - Достъп до различни секции като "Home", "My account", и т.н.
* **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.
  + **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.

# 8 Заключение и възможно бъдещо развитие

С обединяването на всички компоненти в тази документация, проектът SHS създава иновативна образователна среда, подпомагайки на учениците, в техния образователен процес, и създавайки възможности за взаимно обучение и споделяне на знания.

# 9 Използвани литературни източници и Уеб сайтове

(React Documentation, н.д.)