**AI School Recommendation Platform – Документация на проекта**

**Обща информация**

Проектът има за цел да подпомогне седмокласници в избора на най-подходящо училище, базирайки се на техни предпочитания, способности и интереси. Това се реализира чрез уеб приложение, използващо Python и Flask, като в основата стои AI модел за препоръки.

**Инсталация и стартиране**

1. **Клониране на репозиторито:**

bash

CopyEdit

git clone https://github.com/codingburgas/2425-11-g-pp-student-practices-assignment-team10.git

1. **Инсталиране на зависимостите:**

bash

CopyEdit

pip install -r requirements.txt

1. **Стартиране на приложението:**

bash

CopyEdit

python flaskProject/app.py

**Структура на проекта**

csharp

CopyEdit

project/

├── flaskProject/

│ ├── auth/ # Модул за автентикация

│ ├── form/ # Форми за въвеждане

│ ├── static/ # CSS, JS, изображения

│ ├── templates/ # HTML шаблони

│ ├── app.py # Основно приложение

│ └── config.py # Настройки

├── db\_init.py # Скрипт за инициализация на БД

├── requirements.txt # Зависимости

├── README.md # Информация за проекта

**Схема на базата данни (ERD)**

Базата данни съдържа следните основни таблици:

* User(id, username, email, password)
* StudentForm(id, user\_id, score, interests, preferred\_school)
* Recommendation(id, student\_id, school\_name, probability)

**Забележка:** Визуална диаграма може да бъде създадена чрез [dbdiagram.io](https://dbdiagram.io) или [diagrams.net](https://www.diagrams.net) и включена във финалната документация под името schema.png.

**Интеграция на AI модула**

**Данни**

Данните се събират чрез формуляри, попълвани от ученика: академични резултати, интереси, желано училище.

**Процес**

1. Формулярът събира данните на ученика.
2. Данните се предават на AI модул във Flask backend.
3. AI модулът използва алгоритъм (напр. sklearn класификатор или хардкодирана логика), за да изчисли кое училище е най-подходящо.
4. Резултатите се записват в базата и визуализират в интерфейса.

**Документация в кода (docstrings и коментари)**

Примерен docstring:

python

CopyEdit

class StudentForm(FlaskForm):

"""

Форма за въвеждане на данни от ученици:

- среден успех

- любими предмети

- избрано училище

"""

score = IntegerField('Среден успех', validators=[DataRequired()])

interests = StringField('Интереси', validators=[DataRequired()])

**Добри практики:**

* Всеки клас и функция да имат docstring с описание на:
  + входни параметри
  + връщани стойности
  + предназначение
* Коментари в сложни логически блокове (особено при работа с AI модела и обработката на формуляри)

**Agile User Stories**

Регистрация и Аутентикация

1. Като нов потребител, искам да се регистрирам като ученик или учител, за да имам достъп до системата.
2. Като регистриран потребител, искам да мога да вляза в профила си, за да използвам функционалностите.
3. Като потребител, искам да мога да възстановя забравена парола, за да получа отново достъп.
4. Като потребител, искам да мога да изляза от профила си, за да защитя личната си информация.
5. Като потребител, искам да получа информирано съобщение при грешно въведени данни за вход, за да знам какво да коригирам.

Роли: Ученик и Учител

1. Като учител, искам да имам достъп до специфични форми и анкети за преподаване, за да попълвам подходящата информация.
2. Като ученик, искам да мога да попълвам анкета за оценка на учители, за да дам обратна връзка.
3. Като учител, искам да мога да виждам резултатите от алгоритъма за съвпадение със студенти, за да се ориентирам по-добре.
4. Като ученик, искам да ми се предложи най-подходящ учител, според резултатите от попълнените анкети.
5. Като потребител, искам да не мога да достъпвам чужди форми, които не са предназначени за моята роля.

Анкети и формуляри

1. Като учител, искам да мога да попълня анкета, която описва моя опит и стил на преподаване.
2. Като ученик, искам да мога да попълня анкета за нуждите и предпочитанията си.
3. Като потребител, искам формите да бъдат валидирани, за да не изпратя невалидни или непълни данни.
4. Като потребител, искам да получавам потвърждение при успешно попълнена анкета.
5. Като потребител, искам да виждам грешки при неправилно попълнена форма, за да мога да я коригирам.

ML/AI функционалност (Съчетаване учител-ученик)

1. Като учител, искам алгоритъмът да анализира анкетите на учениците, за да предложи най-подходящите за мен.
2. Като ученик, искам алгоритъмът да анализира учителите, за да ми предложи най-доброто съвпадение.
3. Като потребител, искам да виждам резултата от съвпадението визуално, за по-лесно разбиране.

Профили и коментари

1. Като учител или ученик, искам да имам личен профил, в който да се съхранява моята информация.
2. Като потребител, искам да мога да пиша коментари в профила на друг потребител, за да дам препоръка или обратна връзка.
3. Като потребител, искам да мога да виждам коментари, оставени за мен.
4. Като администратор, искам да мога да изтривам неподходящи коментари, за да поддържам културна среда.

Сигурност и валидация

1. Като потребител, искам личните ми данни да бъдат защитени, включително криптирана парола.
2. Като потребител, искам входните данни да се валидарат, за да се предотвратят грешки и злоупотреби.
3. Като потребител, искам да има защита срещу CSRF атаки, за да бъда спокоен при попълване на форми.
4. Като потребител, искам да получавам ясни съобщения при грешка, за да се ориентирам бързо.

Администраторски панел и управление

1. Като администратор, искам да виждам списък с всички потребители, за да ги управлявам.
2. Като администратор, искам да мога да изтривам потребители, които нарушават правилата.
3. Като администратор, искам да мога да блокирам временно достъпа до профили, при нужда.
4. Като администратор, искам да мога да достъпвам отчет за съвпаденията между ученици и учители, за да анализирам ефективността на системата.

**Sprint логове**

| **Седмица** | **Статус** | **Open** | **Closed** | **Изпълнение** |
| --- | --- | --- | --- | --- |
| 1 | ✅ Завършен | 0 | 6 | 100% |
| 2 | ✅ Завършен | 0 | 4 | 100% |
| 3 | ✅ Завършен | 0 | 3 | 100% |
| 4 | ✅ Завършен | 0 | 2 | 100% |
| 5 | ✅ Завършен | 0 | 7 | 100% |
| 6 | 🕐 В процес | 0 | 0 | 0% |

**Заключение**

Проектът демонстрира стабилна архитектура, фокус върху социална полезност чрез AI, и съобразяване с добри практики от Agile методологията. Документацията е изготвена с цел улесняване на поддръжката и бъдещо развитие.

**Препоръки:**

* Добавяне на unit и integration тестове
* Развитие на AI модела (реално обучение)
* Визуализиране на резултатите чрез графики или dashboard