Preppy

An AI driven exam preparation tool for students in the age between 13-16. Teacher can upload their material. The AI creates flashcards, a mockup exam and gives help as a chatbot. The Tool is webbased.

Installation guide

This repository was created under python 3.12.4.

Python installation

Ensure Python 3.12.4 is installed on your system. For Ubuntu:

```
sudo apt install python3.12.4
```

For MacOS:

```
brew install python@3.12.4
```

I myself used Windows and Visual Studio Code to install python. You can download VSC here:

```
https://code.visualstudio.com/download
```

I also installed the extensions for SQLite Viewer and SQLite by alexcuzz in VSC.

Requirements

Make sure you clone the repository properly. After that create a virtual environment with

```
python -m venv venv
```

Get all the necessary requirements:

```
pip install -r requirements.txt
```

Environment

As for the environment, create a new fine called **.env**. This file should contain the following information:

```
APP="run.py"
ENVIRONMENT="Development"
FLASK_SECRET_KEY="YOUR_FLASK_SECRET_KEY" #random key with characters and
numbers
FLASK_SQLALCHEMY_DATABASE_URI="sqlite:///user.sqlite"
OPENAI="YOUR_OPENAI_API_KEY" #for that you have to create an own API. Please
check the website of Openai for that
SECURITY_PASSWORD_SALT = "YOUR_SECURITY_PASSWORD_SALT" #random key with
characters and numbers
BYTES = 749000
FOLDER = "board\static\pdfs"
```

Database

Next you have to set up the database. For **Windows** use:

```
set FLASK_APP=run.py
$env:FLASK_APP = "run.py"
flask init-db
```

After that, the database should be set up.

Now you are all set up!

Running the Project

To start the website, use

```
python run.py
```

You can now find the website under http://127.0.0.1:5000. Have fun!

Introduction

This tool will help teachers and students within the exam preparation. I will exlain the advatages and usage of this toll as well as the fuctionallity.

Motivation

We are in a situation where es lack of a sufficient amount of teachers while the teachers not having enough time for preparing the classes. This tool gives several advantages for both, students and teachers.

Advantages for teachers

The teacher don't need to create specific exam preparation files. The slides from the course are enough. The teacher gets a anonymous overview of the understanding of the students.

Advantages for students

The students can use the tools to prepare for their exam with spefic content, without creating own flashcards. They can talk to a learning assitant at home, when there is no teacher around.

Comparision to other AI Tools

There are several webtools, that help the students to learn with flashcards and further material.

Quizlet

Quizlet is an online tool for students to create flashcards to learn the course material. Here the students have to create the flashcards by themself. Our model creates the flashcards according to the uploaded material from the teacher automatically.

Note

Note is an online tool where students can create pages containing informations about the course mterial, time schedules and grading systems. The students has to fill in the information by themself.

Functions

teachers perspective

The teacher can create a new class and upload course material which is being used by the AI to create flashcards and a mockup exam. The uploaded material is the basis of the learning asisstent.

students perspective

The student can look into the exam preparation via class code that is given by the teacher. Then they have access to automatically designed flashcards, the raw course material, a automatically created mockup exam and a learning assitent, that helps the student with further questions.

Design

The Design of the application was implemented in Figma by Veronika Smilga, Kimberly Sharp and me during the course LLM in education in the university of Tuebingen during the summer semester 2024. Yin-Ying Li was creating the Flowchart based on the Figma Design.

Design on Figma

The figma prototype can be found here:
https://www.figma.com/proto/1VhbgqEpKvuvH2E6JLZWm0/GradeAI-Website?node-id=8-2&t=gnWKger8cISlvrYW-0&scaling=scale-down&content-scaling=fixed&page-id=0%3A1&starting-point-node-id=8%3A2
The prototype includes following pages:

- main
- about
- teachers pages
    - login
    - signup
    - register
    - profile
    - landing page
    - course material
    - uploading page
    - evaluation page
- students pages
    - login
    - signup
    - register
    - profile
    - landing page
    - course material
    - mockup exam
    - mockup evaluation
    - learning assistent
    - flashcards

The Figma prototype was created by Kimberly Sharp, Veronika Smilga and me. It show the bases for the Flask website of this Github page.

Implementation function on Flask

So far the focus of the first draft is to implement a web-chatbot, that can be accessed for the students, when they log in. Therefore the first implemented pages are the following: main, login, signup, students landingpage and chatbot page. Furthermore a flashcard page, the teachers landingpage and the upload page as well as an about page is installed.

The login and signup works via sqlite. The main programming language is python. For the web application, flask is used, implementating both html and css.

For the login it is important that the student can signup as student and the teacher as teacher. It is not aloud to login as teacher when you are actually a student and vice versa. This is handeled via role placing in the signup-page. Therefore the signup has to be on the main page and not the login page, which differs from the initial mockup in Figma.

For the chatbot the first idea was to use llama 3 with ollama so the server is locally. This can increase the privacy of the application. Furthermore Llama can be used without payment. Unfurtunatelly the fuctionallity was to slow.

Next idea was to use llama 2 with a huggingface finetuned model. Nonetheless the kernel crashed.

Now I am using a openai API with the GPT-3.5.turbo model for the chatbot, the summarization of the uploaded data and the flashcard creation.

- [x] main
- [x] about
- [x] teachers pages
  - [x] login
  - [x] signup
  - [ ] profile
  - [x] landing page
  - [ ] course material
  - [x] uploading page
  - [ ] evaluation page
- [x] students pages
  - [x] login
  - [x] signup
  - [x] profile
  - [x] landing page
  - [ ] course material
  - [ ] mockup exam
  - [ ] mockup evaluation
  - [x] learning assistent
  - [ ] flashcards

The folder are structured as showed:
board

- ├ static
- │ ├ css
- │ │ └ styles.css
- │ ├ images
- │ │ ├ arrow-circle-left.png
- │ │ ├ arrow-circle-up.png
- │ │ ├ Background_header.png
- │ │ ├ Backimage.png
- │ │ ├ favicon-16x16.png
- │ │ ├ house-chimney.png
- │ │ ├ icons8-logout-50.png
- │ │ ├ Login Student.png
- │ │ ├ Logo.png
- │ │ ├ Main Page.png
- │ │ ├ Students Flashcards.png
- │ │ └ Students Landing Page.png
- │ ├ js
- │ │ ├ base.js
- │ │ ├ chatbot.js
- │ │ ├ flashcards.js

- ┃ ┃ ┗ 📜upload.js
- ┣ 🗁templates
- ┃ ┣ 🗁auth
- ┃ ┃ ┣ 📜index.html
- ┃ ┃ ┣ 📜login.html
- ┃ ┃ ┗ 📜signup.html
- ┃ ┣ 🗁errors
- ┃ ┃ ┗ 📜404.html
- ┃ ┣ 🗁pages
- ┃ ┃ ┣ 📜about.html
- ┃ ┃ ┣ 📜home.html
- ┃ ┃ ┗ 📜profile.html
- ┃ ┣ 🗁student
- ┃ ┃ ┣ 📜chatbot.html
- ┃ ┃ ┣ 📜flashcards.html
- ┃ ┃ ┗ 📜landing.html
- ┃ ┣ 🗁teacher
- ┃ ┃ ┣ 📜landing.html
- ┃ ┃ ┗ 📜upload.html
- ┃ ┣ 📜base.html
- ┣ 📜auth.py
- ┣ 📜database.py
- ┣ 📜errors.py
- ┣ 📜models.py
- ┣ 📜pages.py
- ┣ 📜schema.sql
- ┣ 📜student.py
- ┣ 📜teacher.py
- ┗ 📜__init__.py
- ┣ 📜.env
- ┣ 📜.gitignore
- ┣ 📜README.md
- ┣ 📜requirements.txt
- ┗ 📜run.py

Limitations

The limit of tokens is quite small. Therefore the uploaded data has to be summarized strongly. A better way to handle that is the RAG LLM which is not used here because of the extensive use of GPU that I could not provide.

Future Work

For the future work, the rest of the Figma Model should be created. This includes the Mockup exam als well as the evaluation for the teacher and the students. Furthermore the profile editing and the coruse material should be created for both student and teacher.

Besides that, the course system has to be installed. students should only get access to there courses. So they should get a second login with the classcode, they want and the opportunity to change the course.

Lastely in future work RAG LLM should be considered because of the big amount of special data that needs to be passed through.

Bibliography

- https://medium.com/@AlexanderObregon/how-to-build-a-simple-chatbot-with-python-4ce0742546a1
- https://pythonspot.com/login-authentication-with-flask/
- https://realpython.com/flask-project/
- https://www.digitalocean.com/community/tutorials/how-to-add-authentication-to-your-app-with-flask-login
- https://medium.com/@abed63/flask-application-with-openai-chatgpt-integration-tutorial-958588ac6bdf
- https://pythonbasics.org/flask-upload-file/
- ChatGPT Use:
    - creating the HTML and CSS componentsfrom the Figma input (manually finegraded)
    - flashcards python and javascript (additioan lpart. The focus of the paper is the chatbot)
    - chatbot error handeling