

MAHATMA EDUCATION SOCIETY'S

PILLAI COLLEGE OF ARTS, COMMERCE & SCIENCE

(Autonomous)

NEW PANVEL

PROJECT REPORT ON

“Course Recommendation System”

IN PARTIAL FULFILLMENT OF

BACHELOR OF COMPUTER SCIENCE

SEMESTER V– 2023-24

PROJECT GUIDE

Prof. Sanjana Bhanagale

SUBMITTED BY: Pawankumar Ghanshyam Jaiswal

ROLL NO: 9127

Course Recommendation System

Project Overview

The Course Recommendation System is a web-based application designed to provide personalized course recommendations to users based on their inputs such as course title, difficulty level, and user preferences. This project utilizes Python's Flask framework for the

backend, a CSV file for managing course data, and HTML/CSS for the frontend.

The system enables users to search for courses, receive tailored recommendations, and view detailed information about suggested courses. User interactions, such as course searches and recommendations, are handled through intuitive web forms and dynamic content updates.

Features

User Interaction

- **Course Search and Recommendation:**

- Users can search for courses by entering a course title and selecting a difficulty level.
- The system retrieves and displays recommendations from a dataset (cleaned_courses_data.csv) based on the provided criteria.
- Recommendations are dynamically generated and presented to the user in an organized manner.

Course Details

- **Course Information Display:**

- Detailed information about each recommended course is shown, including the course title, organization, certificate type, rating, difficulty level, and number of students enrolled.

User Interface

- **User-Friendly Search Form:**

- A responsive search form allows users to input their course preferences and submit queries.
- Suggestions for course titles are provided as users type, enhancing the search experience.

- **Recommendations Layout:**

- Recommendations are displayed in a clear and visually appealing format, including essential details to help users make informed decisions.

- **Course Icon:**

- A distinctive course icon (pj-course-icon.png) is prominently displayed on the user interface to enhance branding and user recognition.

Technical Details

- **Backend:**

- Built using Python's Flask framework, which handles routing, data processing, and dynamic content generation.
- Data is managed and queried from a CSV file (cleaned_courses_data.csv) for recommendations.

- **Frontend:**

- HTML and CSS are used to create a clean and responsive user interface.
- JavaScript handles asynchronous data fetching and dynamic updates to the recommendations section.

• **Functionality:**

- **Search Suggestions:** Provides real-time course title suggestions as users type in the search field.
- **Dynamic Recommendations:** Fetches and displays course recommendations based on user input and selected difficulty level.
- **Loading Indicators:** Shows loading indicators while fetching data to enhance user experience.

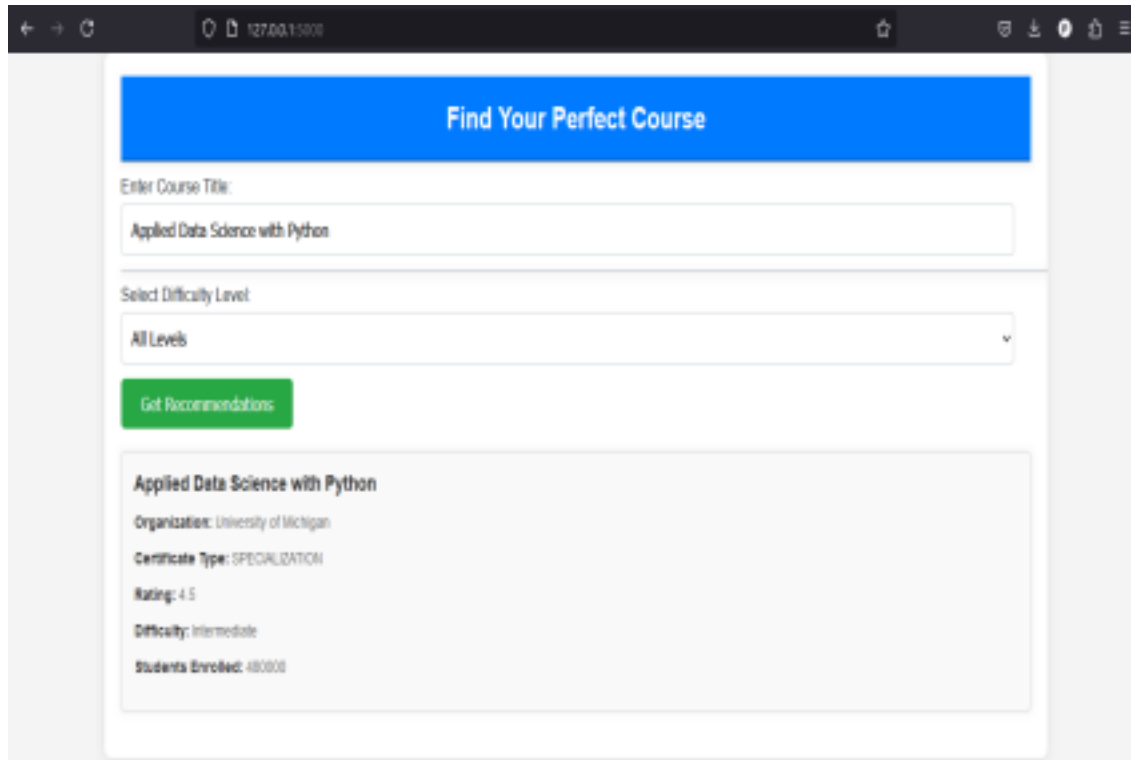
Integration

• **Course Data Management:**

- Utilizes CSV files for storing and managing course data, which is loaded and queried as needed.
- Recommendations are updated based on the data fetched from the CSV file.

```
(.venv) PS C:\Users\pawan\PycharmProjects\Course> python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production ASGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 589-144-587
```

The screenshot displays a web browser window with the address bar showing '127.0.0.1:5000'. The webpage features a blue header with the text 'Find Your Perfect Course'. Below the header, there is a form with two input fields: 'Enter Course Title:' and 'Select Difficulty Level:'. The 'Select Difficulty Level:' dropdown menu is open, showing 'All Levels'. Below the form is a green button labeled 'Get Recommendations'.



Conclusion

The **Course Recommendation System** offers a highly functional, user-friendly platform for users to search and find personalized course recommendations based on their preferences. The use of Flask for backend development ensures that the system remains scalable, while the integration of a dynamic search feature and real-time suggestions enhances the user experience. By incorporating intuitive design elements, the system simplifies the process of exploring online courses. Overall, it provides a comprehensive solution for users seeking tailored course recommendations, ensuring easy access to relevant learning resources.

SWOT Analysis

Strengths

- **Personalization:** The system tailors course recommendations based on user input such as title and difficulty, improving relevance.
- **Simple and Intuitive UI:** A clean, responsive user interface makes the system easy to use for individuals with varying technical abilities.
- **Real-Time Suggestions:** The search field offers real-time course title suggestions, enhancing the user experience.
- **Efficient Data Handling:** Using CSV files for course data ensures a lightweight solution for storing and querying course information.

Weaknesses

- **Limited Data Source:** The system currently relies on a CSV file, which may limit scalability when dealing with large datasets or frequent updates.
- **Lack of User Profiles:** Users do not have personal accounts to save their preferences or history, which could be a useful feature for long-term engagement.
- **Basic Search Criteria:** Search criteria are limited to course title and difficulty. Expanding filters (e.g., course duration, price) would make the recommendations

more comprehensive.

Opportunities

- **Scalability with Database Integration:** Migrating to a relational database like MySQL would allow for larger datasets and more robust querying.
- **User Accounts and Profiles:** Adding user registration and login functionality would allow for personalized recommendations and history tracking.
- **Enhanced Filters and Customization:** Introducing more advanced filters (e.g., categories, reviews, certifications) could enhance the recommendation quality.
- **Integration with Course Platforms:** Partnering with online learning platforms (e.g., Coursera, Udemy) could provide real-time data for recommendations and improve the system's reach.

Threats

- **Competition:** Many established platforms like Coursera and Udemy already provide sophisticated course recommendation engines, making it challenging to compete.
- **Data Privacy Concerns:** Handling personal data, especially in future expansions involving user profiles, could introduce privacy challenges and potential legal obligations.
- **Technological Shifts:** Evolving technologies or the need for more advanced machine learning-based recommendation systems could render the current model outdated.