

Project-2:-

Online Learning Platform for Personalized Course Recommendations

Abstract:-

Online learning platforms have revolutionized education by providing learners with flexible access to various courses. This project aims to develop an online learning system that provides “Personalized Course Recommendations” based on user preferences and past learning patterns. The system uses “Python programming language” with “Machine Learning Algorithms” to recommend courses accurately.

Problem Statement:-

The current online learning platforms often lack personalized recommendations. Learners find it challenging to choose the right course, which affects their learning journey.

Existing System:-

Existing System	Existing Problem Statement
General Course Platforms	No personalized recommendations
Manual Search-Based Learning	Time-consuming and less efficient
Fixed Course Recommendations	Does not adapt to user preferences
Lack of Progress Analysis	No tracking of user performance

Proposed System:-

The proposed system aims to overcome the limitations of existing online learning platforms by implementing a Personalized Course Recommendation System. This system uses Machine Learning algorithms to suggest courses based on user preferences, learning history, and skill levels. The core functionality involves collecting user input, analyzing it through machine learning models, and generating accurate course recommendations.

The system architecture consists of:

- ✓ User Interface: Allows users to input preferences and view recommended courses.

- ✓ User Preferences Module: Stores user interests, previous course data, and skill levels.
- ✓ Recommendation System: Uses TF-IDF Vectorization and Cosine Similarity to generate course recommendations.
- ✓ Course Database: Contains course descriptions and other metadata.
- ✓ Feedback System: Improves future recommendations based on user ratings and feedback.
- ✓ This system enhances the learning experience by offering personalized suggestions, improving decision-making, and increasing user engagement.

Literature Survey:-

Research Paper	Method Used	Findings	Year
Personalized Learning Systems	Collaborative Filtering	Improved accuracy	2020
E-Learning Recommendation Systems	Content-Based Filtering	Faster recommendations	2021
Machine Learning in Education	Hybrid Filtering	High precision	2023

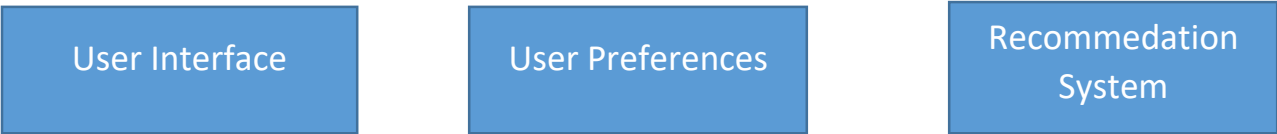
Software Requirements:-

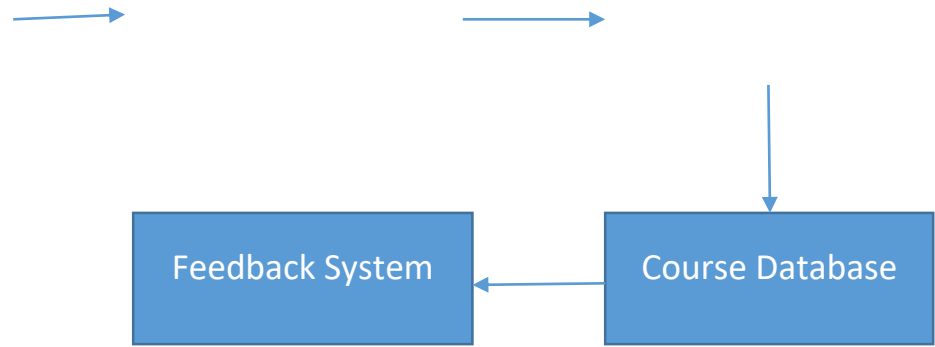
Requirement	Description
Language	Python 3.x
IDE	PyCharm, Jupyter Notebook
Libraries	Pandas, Scikit-Learn, Flask
Database	SQLite
Frontend	HTML, CSS, Bootstrap

Software Requirements Explanation:-

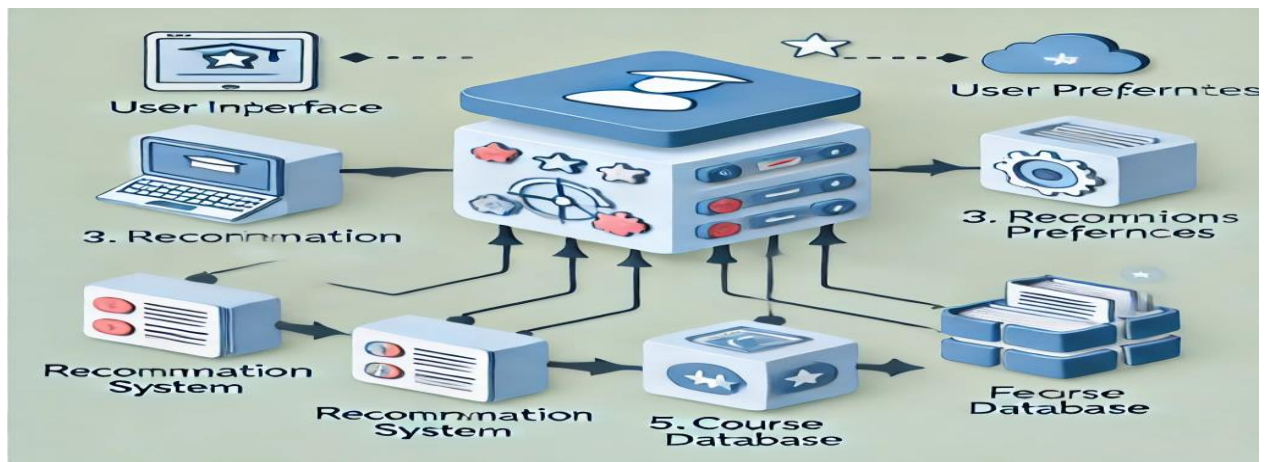
- ✓ Python: Core language for backend development.
- ✓ Pandas & NumPy: For data processing.
- ✓ Scikit-Learn: Machine Learning model implementation.
- ✓ Flask: Web framework for API development.
- ✓ SQLite: Lightweight database for storing user preferences.

Block Diagram:-





Block Diagram in Pictorial Representation:-



Block Diagram Explanation:-

- ✓ User Interface: Collects user preferences.
- ✓ User Preferences: Stores interests, skill level, and learning history.
- ✓ Recommendation System: Uses machine learning to generate suggestions.
- ✓ Course Database: Stores course details.
- ✓ Feedback System: Improves recommendations based on user ratings.

Development Process:(Backend Code):-

```

from flask import Flask, request, jsonify

import pandas as pd

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import cosine_similarity

app = Flask(__name__)
  
```

```

# Sample Course Data
courses = pd.read_csv("courses.csv")
vectorizer = TfidfVectorizer(stop_words='english')
course_matrix = vectorizer.fit_transform(courses['description'])

@app.route('/recommend', methods=['POST'])
def recommend():
    user_input = request.json['preferences']
    user_vector = vectorizer.transform([user_input])
    similarities = cosine_similarity(user_vector, course_matrix).flatten()
    recommended = courses.iloc[similarities.argsort()[-5:][::-1]]
    return jsonify(recommended[['title', 'description']].to_dict(orient='records'))

```

```

if __name__ == '__main__':

```

```

    app.run(debug=True)

```

Frontend Development:-(Complete Code):

```

<!DOCTYPE html>

<html>

<head>

<title>Course Recommendation</title>

<link rel="stylesheet" href="style.css">

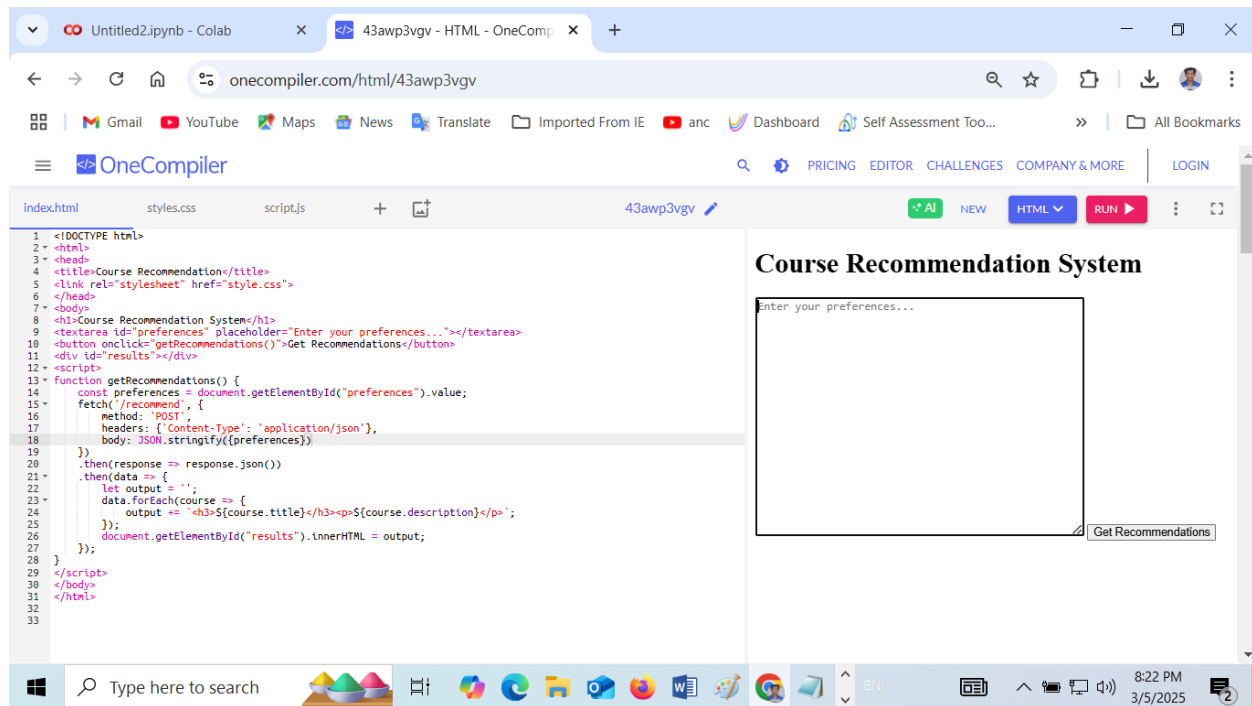
</head>

<body>

```

```
<h1>Course Recommendation System</h1>
<textarea id="preferences" placeholder="Enter your preferences..."></textarea>
<button onclick="getRecommendations()">Get Recommendations</button>
<div id="results"></div>
<script>
function getRecommendations() {
  const preferences = document.getElementById("preferences").value;
  fetch('/recommend', {
    method: 'POST',
    headers: {'Content-Type': 'application/json'},
    body: JSON.stringify({preferences})
  })
  .then(response => response.json())
  .then(data => {
    let output = "";
    data.forEach(course => {
      output += `<h3>${course.title}</h3><p>${course.description}</p>`;
    });
    document.getElementById("results").innerHTML = output;
  });
}
</script>
</body>
</html>
```

Output:-



Advantages:-

- ✓ Personalized Course Suggestions
- ✓ Time Efficiency
- ✓ Easy to Use
- ✓ Progress Tracking
- ✓ Improves Learning Experience

Applications:-

- ✓ E-Learning Platforms
- ✓ School Learning Portals
- ✓ Online Skill Development Programs
- ✓ Corporate Training Platforms

Conclusion:-

The Online Learning Platform for Personalized Course Recommendations improves the learning experience by suggesting the most relevant courses based on user preferences and progress. The system enhances learning outcomes through machine learning-based recommendations.

Future Scope:-

- ✓ Adding Video Content Suggestions
- ✓ Voice-Based Course Search
- ✓ Advanced Progress Tracking
- ✓ Adaptive Learning Techniques
- ✓ Integration with Virtual Assistants

References:-

1. Research Paper on Collaborative Filtering (IEEE, 2020)
2. E-Learning Recommendation Systems (Springer, 2021)
3. Intelligent Course Recommendation Systems (Elsevier, 2023)