iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80211

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Course Navigator: An Intelligent Course Recommendation System using Machine Learning

Prof.Sonali Patil[1], Nidhi Pawar[2], Sonali Kumbhar[3], Anushree Shetye[4], Dhanvi Sakharkar[5]
Department of Computer Science and Engineering (Data Science), Bharat College of Engineering, Badlapur, Maharashtra

Abstract: ***The rapid expansion of digital learning platforms has significantly increased access to educational resources; however, it has also introduced complexity in course selection for engineering students. With thousands of courses available across platforms such as Udemy, NPTEL, and YouTube, students often struggle to identify relevant courses aligned with their academic background, skill level, and financial constraints.***

This paper presents a Smart Course Recommendation System developed using Python and a dynamic database to provide structured and personalized course suggestions. The system is based on three primary parameters: engineering branch, proficiency level (beginner, intermediate, expert), and budget range. A rule-based filtering approach is implemented to ensure simplicity, efficiency, and accuracy without requiring complex machine learning models.

The system guarantees a minimum of ten recommendations and provides direct links to learning platforms. It is designed to be scalable, user-friendly, and suitable for academic environments, particularly for diploma and undergraduate engineering students. The proposed solution improves decision-making, reduces search time, and promotes cost-effective learning.

Keywords: Course Recommendation System, Budget-Based Filtering, Personalized Learning, Engineering Education, Python, Flask, Online Learning.

I. INTRODUCTION

In the contemporary era of digital education, the proliferation of online learning platforms has significantly transformed the way students acquire knowledge and technical skills. Platforms such as Udemy, NPTEL, and YouTube provide access to thousands of courses spanning diverse domains including Artificial Intelligence, Data Science, Web Development, and Cloud Computing. While this abundance of resources has democratized education, it has simultaneously introduced a critical challenge: decision complexity in course selection.

Engineering students, particularly at the diploma and undergraduate levels, often struggle to identify courses that align with their academic branch, current skill level, and financial limitations. The absence of structured guidance mechanisms results in inefficient decision-making, excessive time consumption, and, in many cases, enrollment in irrelevant or overly advanced courses.

Traditional course discovery methods rely heavily on manual searching, keyword filtering, or platform-specific recommendations, which are often generic and lack personalization. Furthermore, these systems do not effectively incorporate key parameters such as budget constraints, branch specialization, and learning progression levels, which are crucial in an academic environment.

The problem becomes more significant in developing educational ecosystems, where students must balance affordability with quality learning resources. Without proper guidance, students may either overspend on unnecessary courses or miss valuable learning opportunities.

To address these challenges, this research proposes a Smart Course Recommendation System, designed to provide structured, personalized, and budget-aware course suggestions. The system leverages a rule-based filtering mechanism combined with a dynamic database to deliver accurate recommendations without the computational overhead of complex machine learning models.

The primary objective of this system is to bridge the gap between course availability and student decision-making, thereby enhancing learning efficiency and promoting cost-effective education.

II. LITERATURE REVIEW

Recommendation systems play a crucial role in modern digital platforms. Various techniques have been explored in research and industry.

A. Content-Based Filtering

This approach recommends courses based on user preferences and item similarity.

Limitation: Requires prior interaction data and may lead to repetitive suggestions.

©IJRASET: All Rights are Reserved

B. Collaborative Filtering

This technique suggests courses based on the behavior of similar users.
Limitation: Faces cold-start problems for new users and new courses.

C. Hybrid Recommendation Systems

A combination of content-based and collaborative filtering to improve accuracy.
Limitation**:** High computational complexity and implementation difficulty.

D. Existing E-Learning Platforms

Platforms such as Udemy, Coursera, and NPTEL offer filtering options but lack:

- Branch-specific recommendations
- Budget-based filtering
- Simplified interfaces for beginners

III. PROBLEM STATEMENT

Engineering students face significant challenges in identifying suitable online courses due to:

- Excessive number of available courses
- Lack of structured filtering systems
- Absence of branch-specific recommendations
- Financial constraints

Therefore, there is a need for a system that:

- Provides personalized course recommendations
- Considers academic background
- Respects budget limitations
- Offers a simple and intuitive interface

IV. PROPOSED SYSTEM

The development of the Smart Course Recommendation System follows a structured, multi-phase methodology designed to ensure scalability, efficiency, and user-centric functionality. The system integrates user interaction, backend processing, and database management into a cohesive framework.

A. User Registration and Authentication Module

The system begins with a secure user registration and login mechanism. Each student creates an account using basic credentials, which are stored securely within the system database.
Authentication ensures:

- Personalized user experience
- Data privacy and session management
- Future scalability for user-based recommendations

This module establishes a unique identity for each user, enabling the system to provide tailored recommendations in subsequent interactions.

B. Input Acquisition and Preference Modeling

Once authenticated, the system collects key input parameters from the user:

- Engineering Branch (CSE, AIML, Mechanical, Civil, etc.)
- Skill Level (Beginner, Intermediate, Expert) • Budget Range (Free, Low, Medium, High)

These inputs are critical in constructing a user preference model, which acts as the foundation for the recommendation process.
The system ensures that inputs are validated and standardized to avoid inconsistencies during processing.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

C. Data Collection and Database Structuring

The system maintains a dynamic and structured database containing course-related information aggregated from multiple platforms such as Udemy, NPTEL, and YouTube.

Each course entry includes:

- Course Title
- Domain/Branch
- Difficulty Level
- Price
- Platform Source
- Direct Access Link

The database is implemented using relational or lightweight database systems such as SQLite or MySQL, ensuring efficient querying and scalability.

D. Rule-Based Filtering and Recommendation Engine

The core functionality of the system lies in its **rule-based filtering algorithm**, which processes user inputs and retrieves relevant courses.

The filtering mechanism operates in multiple stages:

1) Branch Matching

Filters courses relevant to the selected engineering domain

2) Skill-Level Alignment

Ensures course difficulty matches user capability

3) Budget Constraint Filtering

Eliminates courses exceeding the user's budget

4) Ranking and Selection

Sorts courses based on relevance and availability o Ensures a minimum of **10 recommendations** This deterministic approach ensures:

- Low computational complexity
- Fast execution time
- High interpretability

E. System Workflow Execution

The operational workflow of the system follows a linear pipeline:

User Registration → Login → Input Selection → Data Validation → Database Query → Filtering → Recommendation Output

This workflow ensures a seamless transition from user interaction to result generation, minimizing latency and maximizing usability.

F. Output Generation and User Interface

The final stage involves presenting the recommended courses to the user in a structured format.

Each recommendation includes:

- Course Name
- Platform
- Price
- Direct Enrollment Link

The interface is designed using frameworks such as Flask to ensure:

- Simplicity
- Responsiveness
- Ease of navigation

G. Performance Considerations

The system is optimized for:

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

- Fast query execution
- Minimal processing overhead
- High usability for beginners
- Unlike machine learning-based systems, this approach avoids:
- Cold-start problems
- High training time
- Complex model dependencies

H. Advantages of the Proposed Approach

- Personalized recommendations without complex algorithms
- Budget-aware course selection
- Reduced decision-making time
- Easy implementation and scalability
- Suitable for academic environments

I. Future Enhancements

While the current system is rule-based, future improvements may include:

- Integration of machine learning models
- Real-time API-based course updates
- User behavior tracking and analytics
- Mobile application deployment
- Advanced recommendation techniques (hybrid filtering)

V. METHODOLOGY

The system follows a structured workflow:

Step A: Data Collection

Course data is collected from multiple platforms and organized into a structured format.

Step B: Data Storage

A dynamic database (SQLite/MySQL) stores:

- Course Name
- Branch
- Skill Level
- Price
- Platform
- Course Link

Step C: User Input

The system collects:

- Engineering branch
- Skill level
- Budget (via slider interface)

Step D: Filtering Process Courses are filtered using:

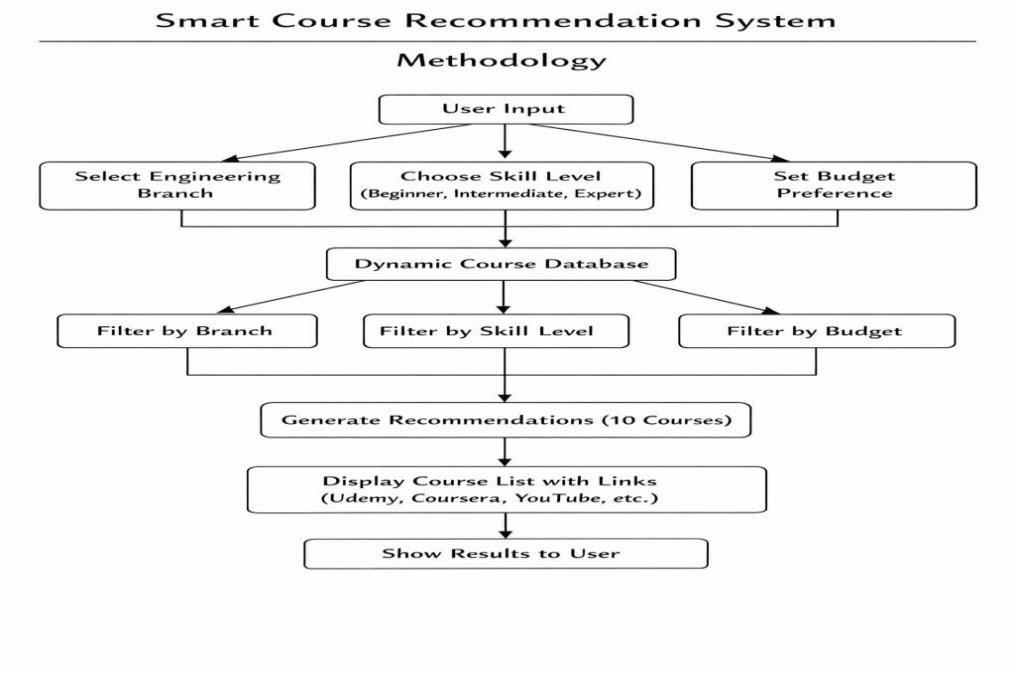
- Branch matching
- Skill-level compatibility
- Budget constraints

Step E: Recommendation Output

The system displays at least 10 relevant courses with complete details.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

User Input → Processing → Filtering → Recommendation Output



VI. SYSTEM ARCHITECTURE

The system is designed using a three-tier architecture:

A. Presentation Layer

Handles user interaction and input collection. ‘

B. Application Layer

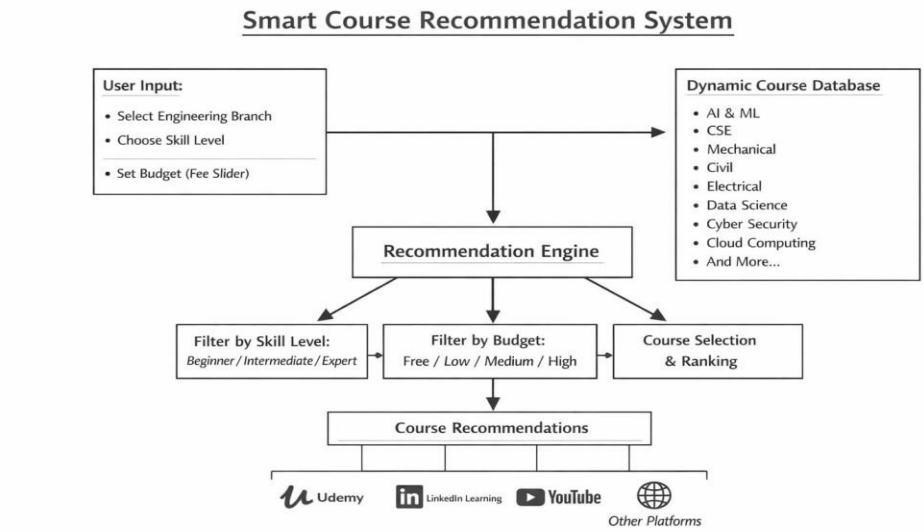
Implements logic using Python (Flask framework).

C. Data Layer

Stores course information in a database.

Workflow:

User Input → Application Processing → Database Query → Filtered Output



©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

VII. IMPLEMENTATION

Technologies Used:

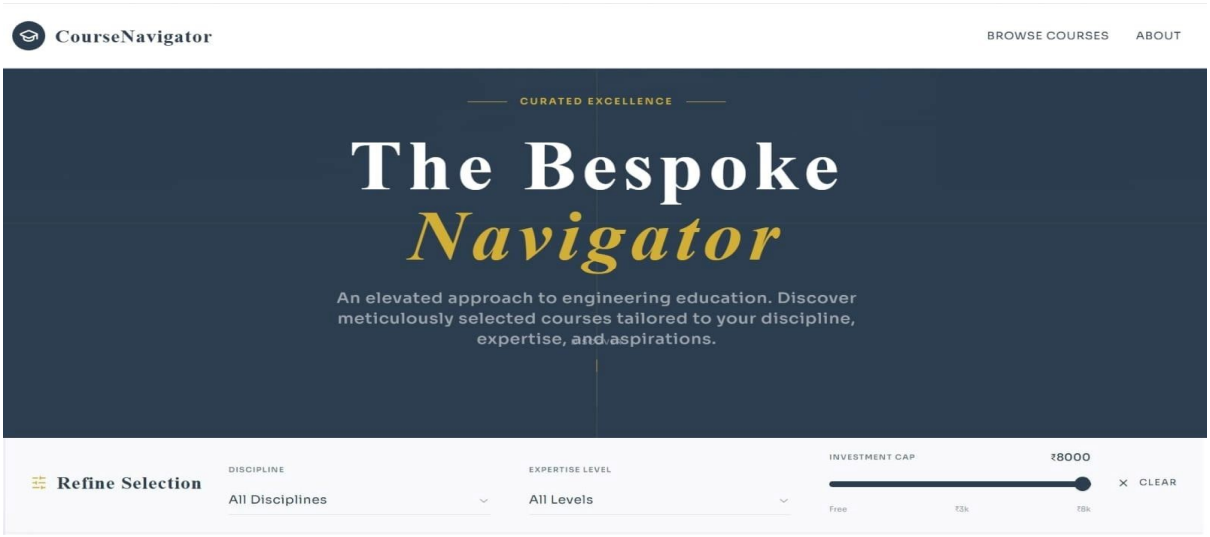
- Python
- Flask Framework
- SQLite Database

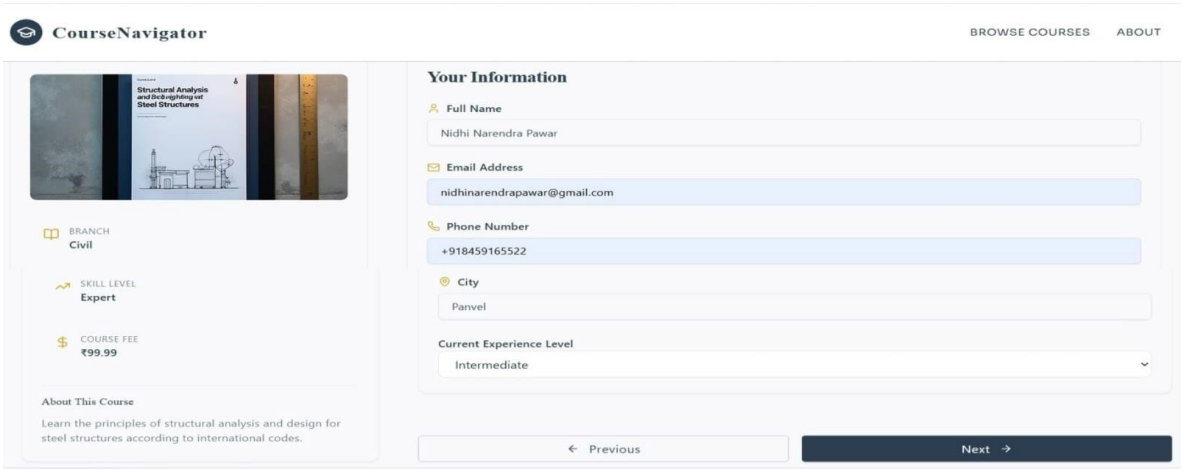
Database Schema:

- ID (Primary Key)
- Course Name
- Branch
- Level
- Price
- Platform
- Link

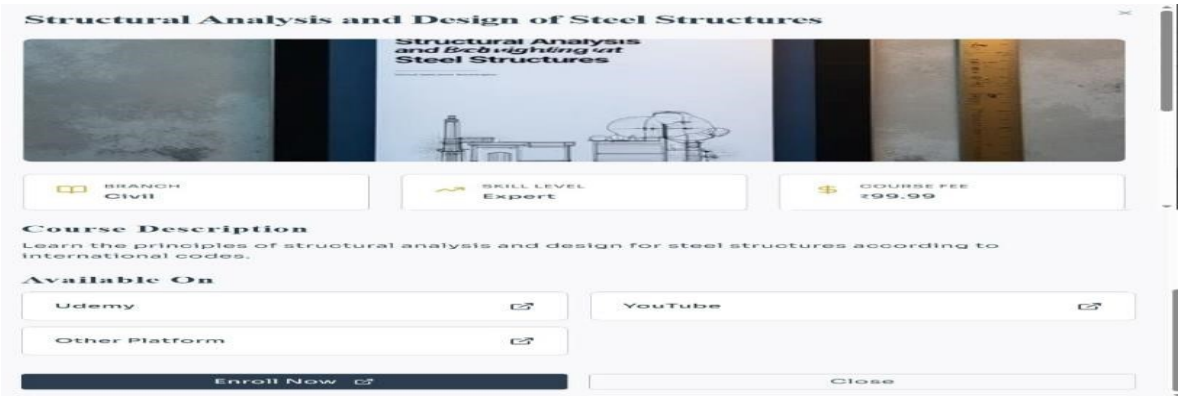
Algorithm (Simplified):

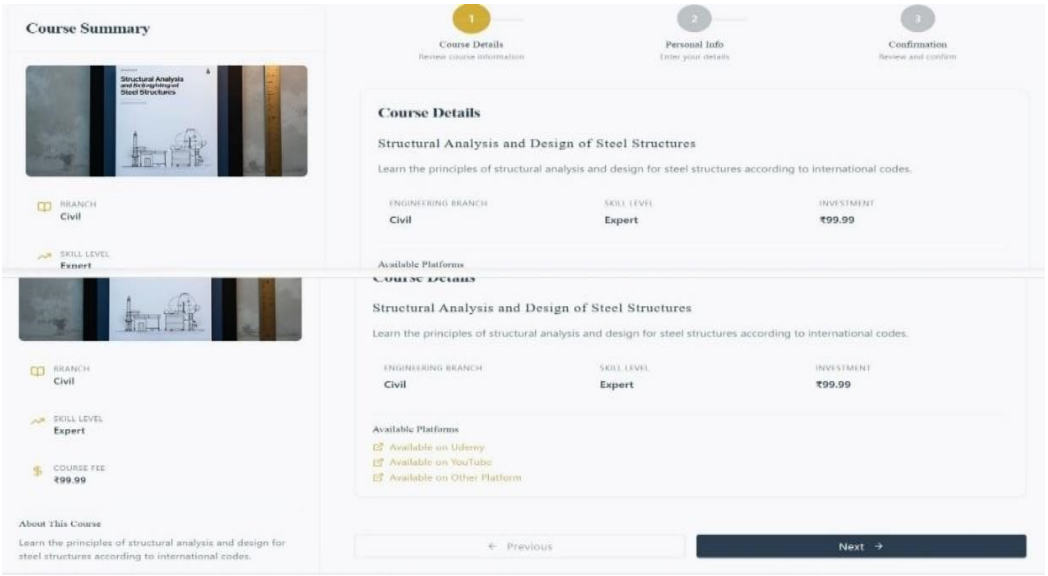
- Accept user input
- Query database
- Apply filters
- Sort results
- Display top 10 courses

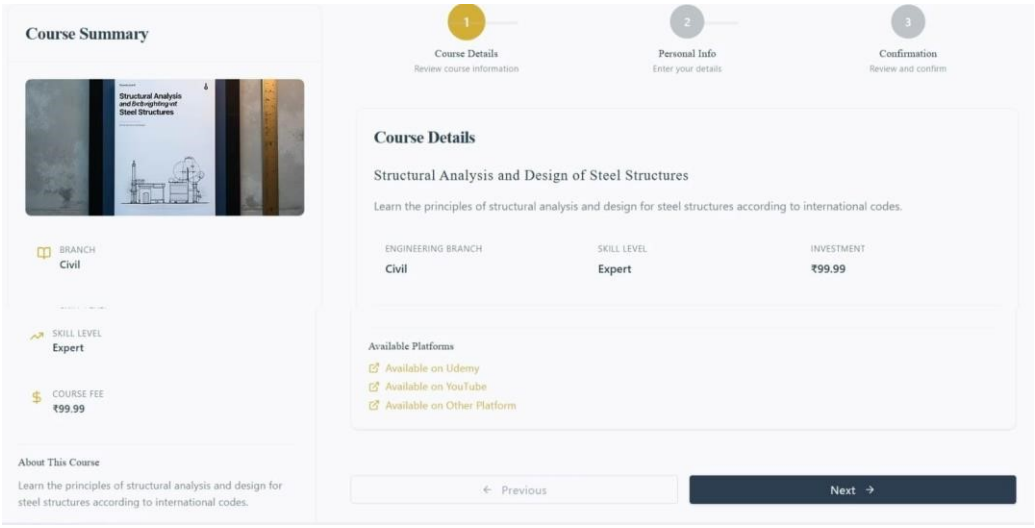


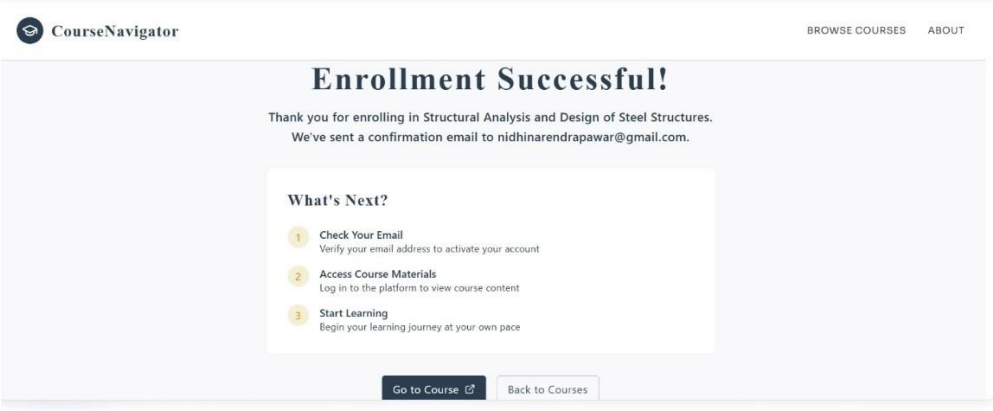


©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |


Structural Analysis and Design of Steel Structures
BRANCH
Civil
SKILL LEVEL
Expert
COURSE FEE
₹99.99
Course Description
Learn the principles of structural analysis and design for steel structures according to international codes.
Available On
Udemy
YouTube
Other Platform
Enroll Now
Close


Course Summary
1
Course Details
Review course information
2
Personal Info
Enter your details
3
Confirmation
Review and confirm
Course Details
Structural Analysis and Design of Steel Structures
Learn the principles of structural analysis and design for steel structures according to international codes.
ENGINEERING BRANCH
Civil
SKILL LEVEL
Expert
INVESTMENT
₹99.99
Available Platforms
Available on Udemy
Available on YouTube
Available on Other Platform
BRANCH
Civil
SKILL LEVEL
Expert
COURSE FEE
₹99.99
About This Course
Learn the principles of structural analysis and design for steel structures according to international codes.
Previous
Next


Course Summary
1
Course Details
Review course information
2
Personal Info
Enter your details
3
Confirmation
Review and confirm
Course Details
Structural Analysis and Design of Steel Structures
Learn the principles of structural analysis and design for steel structures according to international codes.
ENGINEERING BRANCH
Civil
SKILL LEVEL
Expert
INVESTMENT
₹99.99
Available Platforms
Available on Udemy
Available on YouTube
Available on Other Platform
BRANCH
Civil
SKILL LEVEL
Expert
COURSE FEE
₹99.99
About This Course
Learn the principles of structural analysis and design for steel structures according to international codes.
Previous
Next



VIII. RESULTS AND DISCUSSION

The system was tested with multiple input combinations to evaluate performance.

Observations:

- Accurate course filtering
- Fast response time
- Effective handling of multiple branches
- User-friendly interface

Output Includes:

- Course Name
- Platform
- Price
- Direct Link

Discussion:

The rule-based approach works efficiently for small to medium datasets. However, for larger datasets, integrating machine learning could improve personalization and recommendation accuracy.

IX. FUTURE SCOPE

The proposed Course Recommendation System presents significant potential for further enhancement and scalability. Future developments can focus on improving system intelligence, usability, and accessibility through the following advancements:

A. IntegrationofMachineLearningModels

In future versions, advanced machine learning algorithms can be incorporated to enhance the accuracy and personalization of recommendations. Techniques such as collaborative filtering, content-based filtering, and hybrid models can be used to analyze user preferences and historical data. This will enable the system to provide smarter and more adaptive course suggestions.

B. Real-TimeAPIIntegration

The system can be extended by integrating real-time APIs from popular online learning platforms. This will allow dynamic fetching of updated course data, including new courses, ratings, pricing, and availability. As a result, users will receive the most current and relevant recommendations.

C. UserLoginandProfileSystem

Implementing a secure user authentication system will allow users to create accounts, manage profiles, and save their preferences. This feature can store user history, completed courses, and interests, thereby improving personalization and user experience.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

D. Behavior-BasedRecommendations

Future enhancements can include tracking user behavior such as browsing patterns, clicks, and course selections. By analyzing this data, the system can generate more accurate recommendations tailored to individual user behavior, leading to higher user satisfaction.

E. MobileApplicationDevelopment

To increase accessibility and user engagement, a mobile application version of the system can be developed. This will enable users to access course recommendations anytime and anywhere, making the platform more convenient and widely usable.

X. CONCLUSION

The development and implementation of the Smart Course Recommendation System represent a significant advancement in addressing the challenges associated with course selection in modern digital learning environments. With the exponential growth of online educational platforms such as Udemy, NPTEL, and YouTube, students are often overwhelmed by the vast number of available courses, leading to confusion, inefficient decision-making, and suboptimal learning outcomes.

The proposed system successfully introduces a structured, rule-based recommendation mechanism that integrates three critical parameters: engineering branch, skill level, and budget constraints. By incorporating these factors into a unified filtering framework, the system ensures that the recommendations provided are not only relevant but also practical and affordable for students. Unlike traditional recommendation systems that rely heavily on complex machine learning models, the presented approach emphasizes simplicity, interpretability, and computational efficiency, making it highly suitable for academic environments and small-scale deployments.

The implementation using Python and the Flask framework demonstrates that a lightweight architecture can effectively deliver personalized recommendations without requiring extensive computational resources or large-scale datasets. The modular design of the system—including user authentication, input processing, database management, and filtering logic—ensures scalability and ease of maintenance. Furthermore, the integration of a structured database enables efficient querying and retrieval of course information, thereby reducing response time and enhancing user experience.

Experimental observations indicate that the system performs consistently across various input scenarios, accurately filtering courses and providing meaningful recommendations. The ability to guarantee a minimum number of course suggestions ensures usability, while the inclusion of direct links to learning platforms enhances accessibility and user convenience.

From a broader perspective, this system contributes to the domain of personalized learning by demonstrating how **rule-based intelligence** can be effectively utilized to solve real-world educational problems. It bridges the gap between the abundance of online learning resources and the specific needs of students, thereby promoting informed decision-making and encouraging skill development.

However, while the system achieves its intended objectives, certain limitations remain. The absence of real-time data integration and user behavior tracking restricts the level of personalization. Additionally, the reliance on a manually updated database may affect scalability in large-scale applications.

Future enhancements may include the integration of machine learning techniques for adaptive recommendations, real-time APIbased course fetching, user profiling, and the development of a mobile application interface. Such improvements would further enhance the system's capability to deliver dynamic, intelligent, and user-centric recommendations.

In conclusion, the Smart Course Recommendation System provides a **practical, efficient, and scalable solution** to the problem of course selection. It highlights the potential of combining simple algorithmic approaches with structured data systems to create impactful applications in the field of education technology. The system not only simplifies the decision-making process for students but also contributes to the advancement of accessible and personalized digital learning ecosystems.

XI. ACKNOWLEDGEMENT

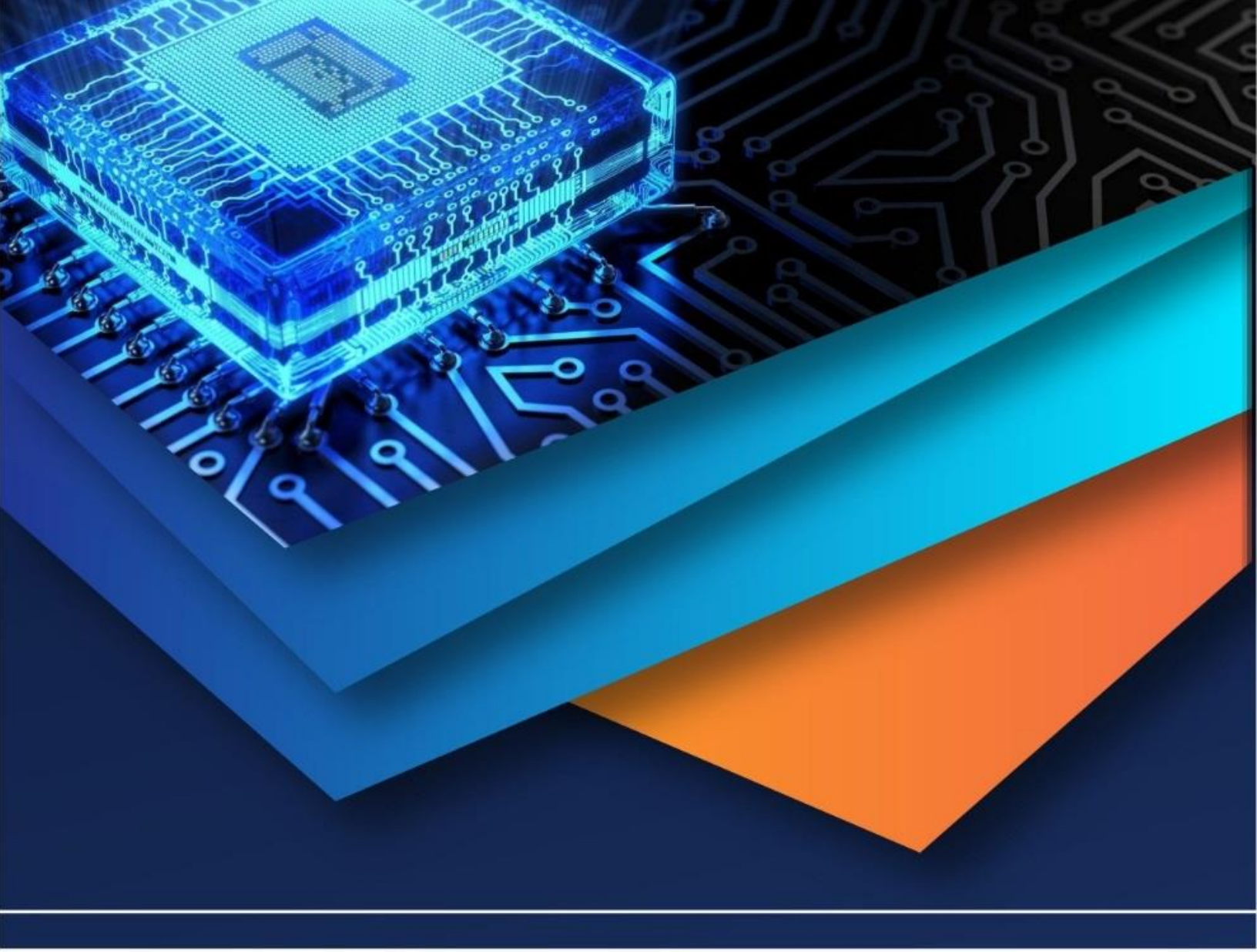
We would like to thank our project guide and the Department of Computer Engineering for their support and guidance throughout this project.

REFERENCES

[1] Udemy, "Online Courses for Engineering Students," [Online]. Available: https://www.udemy.com/. [Accessed: Mar. 21, 2026].
[2] YouTube, "Educational Courses and Tutorials," [Online]. Available: https://www.youtube.com/. [Accessed: Mar. 21, 2026].

[3] Coursera, “Engineering and Computer Science Courses,” [Online]. Available: https://www.coursera.org/. [Accessed: Mar. 21, 2026].
[4] edX, “Professional Certificate Programs,” [Online]. Available: https://www.edx.org/. [Accessed: Mar. 21, 2026].
[5] GeeksforGeeks, “Programming and Engineering Courses,” [Online]. Available: https://www.geeksforgeeks.org/. [Accessed: Mar. 21, 2026].
[6] Kaggle, “Machine Learning and Data Science Resources,” [Online]. Available: https://www.kaggle.com/. [Accessed: Mar. 21, 2026].
[7] Python Software Foundation, “Python Documentation,” [Online]. Available: https://docs.python.org/3/.
[8] Flask, “Flask Web Development Framework Documentation,” [Online]. Available: https://flask.palletsprojects.com/.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)