Synopsis Report On



# STUDENT PERFORMANCE ANALYZER

Submitted in fulfilment of the requirements of the degree of

Bachelor of Engineering by

**Himanshu Bind (10)  
Mahek Gupta (42)  
Akhilesh Patil (54)**

**Devansh Pandey (53)**

Under Supervision of Prof. S. S. Kadam

DEPARTMENT OF INFORMATION TECHNOLOGY BHARATI VIDYAPEETH COLLEGE OF ENGINEERING, NAVI MUMBAI

2024-25

# Synopsis Report Approval For Mini Project



This is to certify that,

**Himanshu Bind (10)**

**Mahek Gupta (42)**

**Akhilesh Patil (54)**

**Devansh Pandey (53)**

have completed the Mini Project STUDENT PERFORMANCE ANALYZER satisfactorily in the Department of Information Technology as prescribed by the Mumbai University in the academic year 2024-2025

Prof. S. S. Kadam Project Guide

Prof. Sonali N. Mhatre Dr. Sandhya Jadhav

Head of Department Principal

Date:- Place:-

**Declaration**



We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Name of student and Roll No.) (Signature)

(Name of student and Roll No.) (Signature)

(Name of student and Roll No.) (Signature)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Name of student and Roll No.) (Signature)

Date:

**Table of Content**



|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Content** | **Page No.** |
|  | List of Figures |  |
|  | List of Tables |  |
| 1. | Introduction | 1 |
| 2. | Literature Review | 2-13 |
| 3. | Methodology | 14-15 |
| 4. | Problem Definition | 16 |
| 5. | Objectives | 17 |
| 6. | Diagrams | 18-20 |
| 7. | Software Requirements | 21 |
| 8. | Implementation | 22-25 |
| 9. | Result & Analysis | 26 |
| 10. | Conclusion | 27 |
| 11. | References | 28 |

**List of Figure**



|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| Fig 6.1. | Use Case Diagram | 18 |
| Fig 6.2. | E-R Diagram | 19 |
| Fig 6.3 | Work Flow Diagram | 20 |
| Fig 8.1 | Welcome page | 23 |
| Fig 8.2 | Front End | 24 |
| Fig 8.3 | Back end | 25 |



**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Title** | **Page No.** |
| 2.1 | Literature Review Papers | 2 |
| 7.1 | Software Requirements | 21 |

**ABSTRACT**



Traditional recruitment is often inefficient and biased, relying heavily on CGPA without considering other essential attributes. **SkillScore AI** is an **AI-driven student recommendation and recruiter filtering system** designed to enhance **placement processes** in academic institutions. It evaluates **CGPA, technical skills, leadership roles, extracurricular activities, and work experience** using **machine learning algorithms** to ensure a **holistic student assessment**.

Unlike conventional systems, SkillScore AI assigns **custom weightages** to various factors, enabling recruiters to **filter, rank, and shortlist candidates objectively**. Built with **Flask, Scikit-Learn, and MySQL**, it offers a **scalable, web-based interface** with **85% accuracy** in career predictions.

By incorporating **data-driven hiring practices**, SkillScore AI fosters **fair and efficient placements**, ensuring well-rounded candidates receive equal opportunities alongside academically strong students, ultimately bridging the gap between students and recruiters.

# CHAPTER I



**INTRODUCTION**

Finding the right job or career path is one of the biggest challenges students face during placements. Traditional hiring methods rely too much on **CGPA**, often ignoring important factors like **technical skills, leadership experience, extracurricular activities, and internships**. This means many talented students miss out on opportunities just because they don’t have the highest grades. **SkillScore AI** is designed to change that by offering a **smart, AI-powered system** that evaluates students based on a **combination of academic and non-academic achievements**.

SkillScore AI uses **machine learning** to analyze a student’s profile holistically. Instead of just looking at CGPA, it considers **technical expertise, projects, work experience, leadership roles, and extracurricular involvement**. Recruiters can also **customize weightages** based on what matters most to their hiring needs. This way, companies get to filter and rank candidates based on **real skills and potential**, not just exam scores.

The system is built using **Flask for backend processing, Scikit-Learn for machine learning predictions, and MySQL for managing student data**. It comes with a **simple, web-based interface** that allows students to get **career recommendations**, while recruiters can easily **search, filter, and shortlist candidates** based on their requirements.

SkillScore AI achieves **85% accuracy** in career predictions and **reduces the time spent on shortlisting candidates**. This makes hiring more **efficient, fair, and data-driven**, ensuring that students with diverse strengths get recognized. Moving forward, the system will be expanded to include **resume parsing, job portal integration, and AI-driven career counseling** to make student hiring even smarter and more effective.

# CHAPTER II LITERATURE REVIEW



## Literature Review Papers



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Paper No.** | **Title** | **Objective** | **Methodology** | **Disadvantage** | **Conclusion** |
| 1. |  |  | Implemented data mining techniques such as Naïve Bayes and K-Nearest Neighbor (KNN) for classification. Conducted a case study on 200 students. |  |  |
|  | **Student Performance Analysis Using Machine**  **Learning**  **AUTHOR**:  Kotagiri Sanjana et al. (2023) | To develop a rule-based recommender system for analyzing and predicting student performance based on demographic, study-related, and psychological characteristics. | The generated rules did not fully extract reasons behind student dropouts. Recommendations lacked clear methodologies for implementation. | The proposed framework successfully identified student weaknesses and provided recommendations for improvement, proving effective when compared to existing models. |



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Paper No.** | **Title** | **Objective** | **Methodology** | **Disadvantage** | **Conclusion** |
| 2. |  |  | Used ML models such as Logistic Regression, Decision Trees, and Support Vector Machines (SVM) with data collected through structured questionnaires. |  |  |
|  | **Student General Performance Prediction Using Machine Learning Algorithm**  **AUTHOR**:  Shikha Pachouly et al. (2022) | To predict student academic performance by incorporating traditional academic metrics as well as psychological and emotional factors. | Emotional and psychological factors were not traditionally included in prediction models, leading to limited accuracy in past studies. | Incorporating non-academic attributes like mental state and personal interests improved prediction accuracy and provided a more holistic view of student performance. |



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Paper No.** | **Title** | **Objective** | **Methodology** | **Disadvantage** | **Conclusion** |
| 3. |  |  | Employed Multivariate Linear Regression, Random Forest, and Gradient Boosting algorithms on datasets containing previous marks, attendance, and parental education levels. |  |  |
|  | **Performance Analysis and Prediction of Student Result Using Machine Learning**  **AUTHOR**:  Prakash, Sachin Garg (2021) | To automate student performance evaluation and predict future results based on academic and socio-economic factors. | The dataset was limited to Portuguese school students, and real-time adaptability was not tested. | ML-based predictive models provided accurate results, allowing faculty to guide students towards better academic outcomes. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Paper No.** | **Title** | **Objective** | **Methodology** | **Disadvantage** | **Conclusion** |
| 4. |  |  | Applied various ML algorithms such as Decision Trees, Naïve Bayes, and Neural Networks. Specific dataset details were not provided. |  |  |
|  | **Predicting Students' Performance Using Machine Learning Algorithms** | To use machine learning techniques to analyze and predict student academic success. | Lack of clarity in dataset sources and methodology details. | ML techniques have been demonstrated as reliable methods for student performance prediction, though more transparency in implementation is needed. |



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Paper No.** | **Title** | **Objective** | **Methodology** | **Disadvantage** | **Conclusion** |
| 5. |  |  | Used Naïve Bayes and K-Nearest Neighbor algorithms on a dataset with five key factors: attendance, assignments, projects, lab performance, and seminars. Statistical analysis tool ‘R’ was used. |  |  |
|  | **Student Performance Analysis Using Machine Learning Tools**  **AUTHOR**:  Atul Prakash Prajapati et al. (2017) | To evaluate and improve student performance using machine learning-based cognitive modeling and statistical analysis. | The Naïve Bayes model had an error ratio of 20:5, limiting the accuracy to 75%. Did not explore deep learning approaches. | The study concluded that cognitive modeling and machine learning tools are effective for student performance analysis, but future work should explore advanced ML techniques for higher accuracy. |



# CHAPTER III METHODOLOGY



**1. Data Collection & Preprocessing**

**1.1 The system gathers student information from a dataset that includes:**

* **Academic Performance – CGPA, subject-wise scores**
* **Technical Skills – Programming languages, certifications**
* **Extracurricular Activities – Leadership roles, competitions, clubs**
* **Internships & Work Experience**

**1.2 Before model training, data preprocessing is performed:**

* **Handling Missing Values – Filling gaps using mean/mode imputation**
* **Encoding Categorical Data – Converting textual information (e.g., leadership roles) into numerical values**
* **Feature Scaling – Normalizing numerical attributes to ensure uniform comparisons**

**2. Machine Learning Model Development**

**2.1 Random Forest Classifier**

**A Random Forest Classifier is trained on labeled student data to predict suitable career paths. The model considers multiple input features and learns patterns from past students' data.**

**2.2 Training the model**



* **The training process includes:**
  + **Splitting data into training and testing sets**
  + **Evaluating model accuracy and tuning hyperparameters**
  + **Saving the trained model as model.pkl for deployment**

**3. Student Ranking System**

**3.1 Holistic evaluation**

**To ensure a holistic evaluation, a ranking mechanism is introduced, considering both academic and extracurricular factors.**

**FinalScore=(0.55×GPA)+(0.15×TechnicalSkills)+(0.12×Internships)+(0.1×LeadershipScore)+(0.08×ExtracurricularScore)**

**3.2 Ranking Process:**

* **Students are assigned scores based on the weighted formula.**
* **d, considering both academic and extracurricular factors.**
* **The system dynamically sorts students in descending order based on their final scores.**
* **Recruiters can apply filters to shortlist candidates based on minimum scores.**

**4. Backend Development (Flask API)**

* **The backend is developed using Flask, a lightweight Python framework.**
* **It loads the trained machine learning model and ranking algorithm.**
* **Flask handles requests from the frontend and returns career recommendations and ranked student lists.**

**5. Frontend Development (HTML, CSS, JavaScript)**



* **The UI is designed using HTML, CSS, and JavaScript for a user-friendly experience.**
* **Forms collect student details, which are sent to the Flask API for processing.**
* **The recruiter dashboard allows filtering based on custom weightages (GPA, skills, experience, extracurriculars).**

**6. Database Management (SQLite/MySQL)**

* **Student data and recruiter preferences are stored in a relational database.**
* **Flask interacts with the database to retrieve and update records dynamically.**

**7. Deployment & Future Enhancements**

* **The application is deployed locally, with future plans for cloud deployment.**
* **Upcoming features include resume parsing with NLP, job portal integration, and AI-based career counseling chat support.**

# CHAPTER IV PROBLEM DEFINITION



The traditional recruitment process in academic institutions primarily relies on **CGPA**, often overlooking **technical skills, leadership abilities, extracurricular achievements, and work experience**. This creates an unfair advantage for students with high grades while ignoring those who excel in practical skills and real-world problem-solving.

Recruiters also face challenges in **efficiently filtering and ranking candidates**, as manual resume screening is **time-consuming and inconsistent**. The lack of a standardized system leads to **biased hiring decisions** and **missed opportunities** for well-rounded students.

**SkillScore AI** aims to solve this issue by introducing an **AI-driven evaluation system** that considers **both academic and non-academic factors**. It allows recruiters to **customize selection criteria**, ensuring that students are assessed based on **job-relevant skills and industry needs** rather than just grades.

By implementing **SkillScore AI**, we aim to **streamline hiring, reduce bias, and create a more effective, skills-based recruitment process** that benefits both students and recruiters.

# CHAPTER V OBJECTIVES



**1. Improve Career Recommendations for Students**

* Provide **personalized career guidance** based on students' **GPA, skills, work experience, and extracurricular activities**.
* Use **machine learning models** to predict the most **suitable career paths**.
* Ensure students receive **realistic and data-driven** job role suggestions.

**2. Implement an Intelligent Student Ranking System**

* Develop a **weighted scoring mechanism** that ranks students based on:
  + **55% CGPA**
  + **15% Technical Skills**
  + **12% Internships**
  + **10% Leadership Roles**
  + **8% Extracurricular Activities**
* Dynamically **sort students** from highest to lowest based on their final score.
* Allow recruiters to filter candidates using **minimum ranking thresholds**.

**3. Enable Data-Driven Shortlisting for Recruiters**

* Provide an **automated system** for **efficient and bias-free hiring**.
* Reduce manual resume screening by using **AI-powered candidate filtering**.
* Allow recruiters to **customize selection criteria** based on job-specific needs.

**4. Develop an Interactive & Scalable Web Application**

* Build a **user-friendly web interface** for both students and recruiters.
* Use **Flask (Python) as the backend** to process student data and return rankings.
* Store data securely in **SQLite/MySQL for dynamic updates and retrieval**.

**5. Deploy & Improve the System for Future Use**

* Ensure the system can be **deployed on the cloud** for broader accessibility.
* Plan future improvements, including:
  + **Resume Parsing using NLP** for automatic skill extraction.
  + **Job Portal Integration** to match students with real-time job openings.
  + **AI Chatbot for Career Counseling** to provide instant career advice.

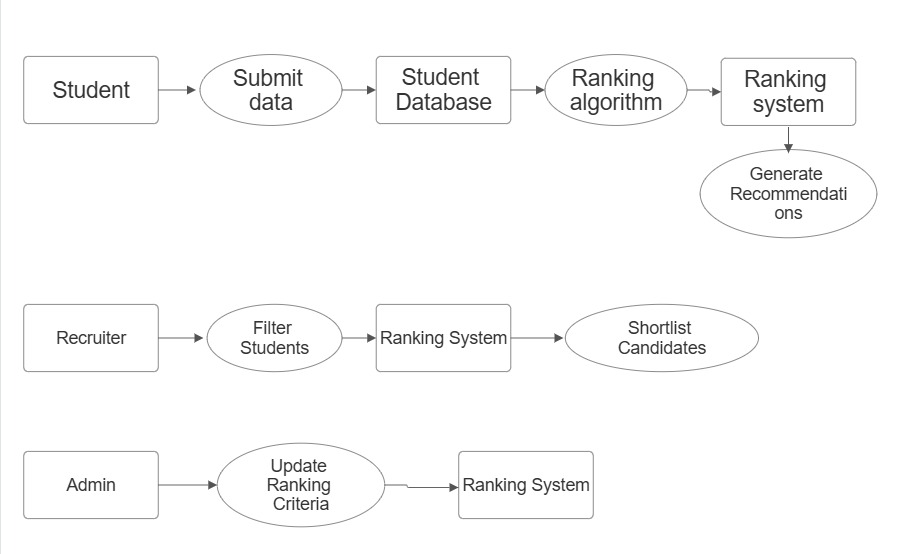
# CHAPTER VI



# DIAGRAMS

# 

*Fig.6.1: Use Case Diagram*



*Fig.6.2 Data Flow Diagram*



# CHAPTER VII SOFTWARE REQUIREMENTS



|  |  |
| --- | --- |
| **Name of the Component** | **Specification** |
| **Operating System** | Windows 10/11, macOS, Linux |
| **Programming Language** | Python 3.8+ |
| **Web Framework** | Flask |
| **Machine Learning** | Scikit-Learn, NumPy, Pandas |
| **Model Algorithm** | Random Forest Classifier |

*Table 7.1. Software Requirements*

# CHAPTER VIII IMPLEMENTATION



The SkillScore AI project was implemented using Machine Learning.

**1️.Set Up the Environment**

* Install Python 3.8+, Flask, Scikit-Learn, NumPy, Pandas, and database dependencies.
* Set up a virtual environment using virtualenv.

**2️.Data Collection & Preprocessing**

* Gather student data (CGPA, skills, internships, leadership roles, extracurriculars).
* Clean and normalize the data to ensure consistency.

**3️.Train the Machine Learning Model**

* Use Random Forest Classifier to predict career recommendations.
* Save the trained model using pickle.

**4️.Develop Backend with Flask**

* Load the trained ML model.
* Create API endpoints for student recommendations and ranking.

**5️.Build the Frontend**

* Develop an HTML, CSS, and JavaScript-based UI for students and recruiters.
* Implement forms for input and display ranked student results.

**6️.Implement Student Ranking System**

* Use a weighted formula to calculate the Final Score based on CGPA, skills, internships, leadership, and extracurriculars.
* Sort students dynamically from highest to lowest score.

**7️.Integrate Database (SQLite/MySQL)**

* Store student data and recruiter preferences.
* Enable dynamic updates and retrieval of student records.

**8️.Testing & Debugging**

* Test model accuracy and adjust weights if needed.
* Fix UI/Backend bugs and optimize query performance.

**9️.Deploy the Web Application**

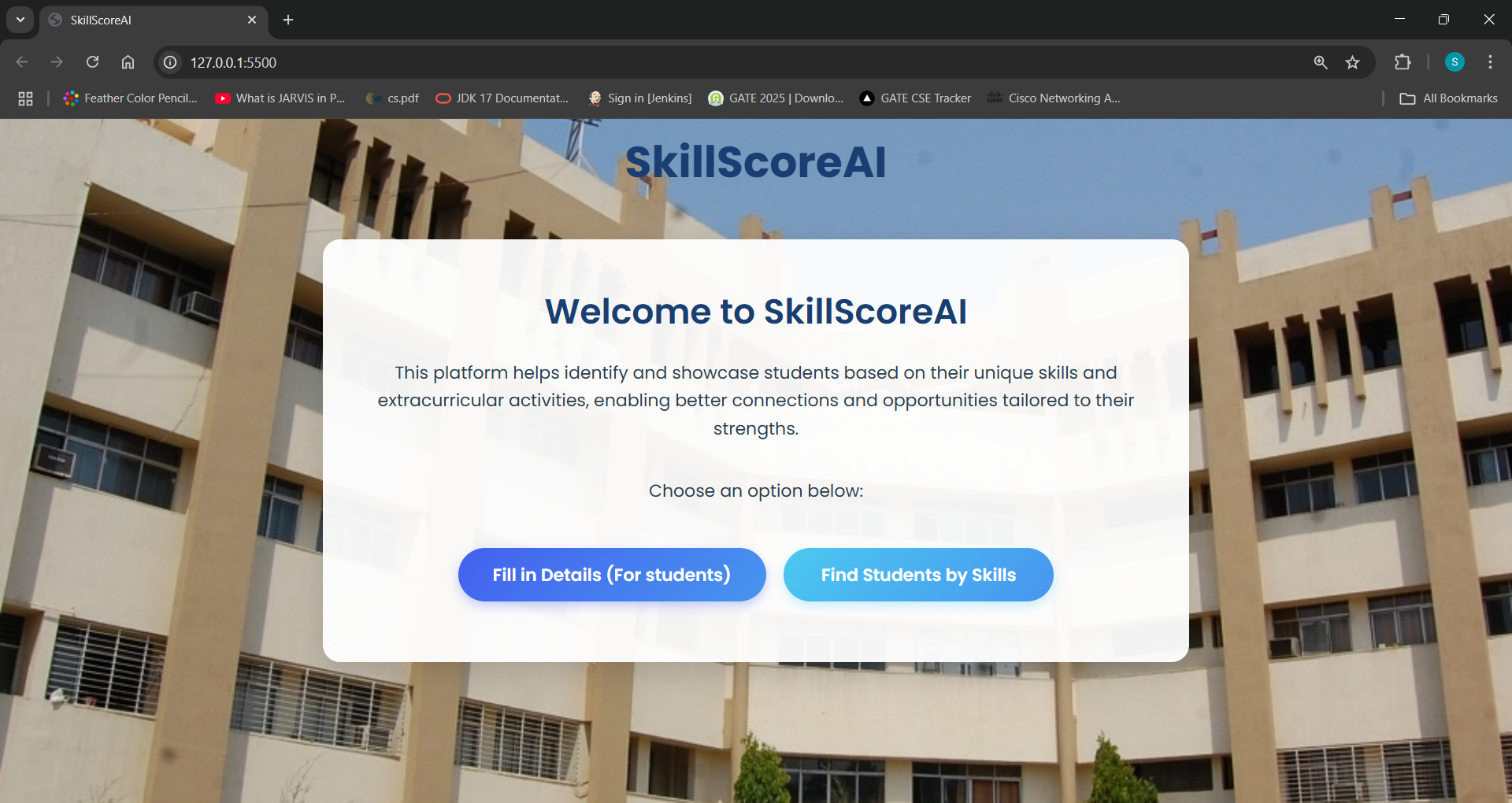
* Run the Flask server locally using python app.py.
* Deploy to AWS/GCP (Future Scope).

**10. Future Enhancements**

* Add resume parsing with NLP.
* Integrate with job portals for real-time placements.

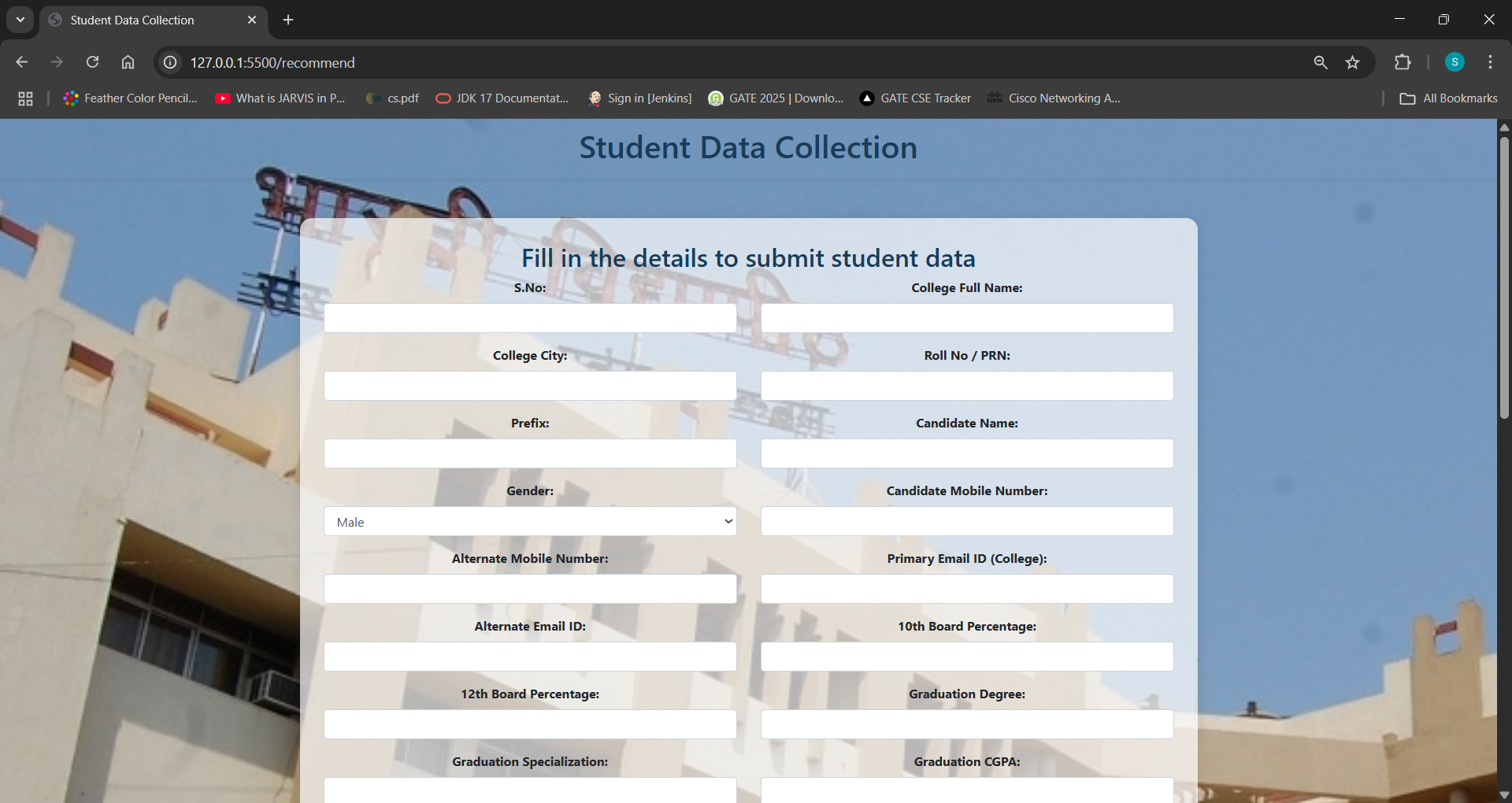




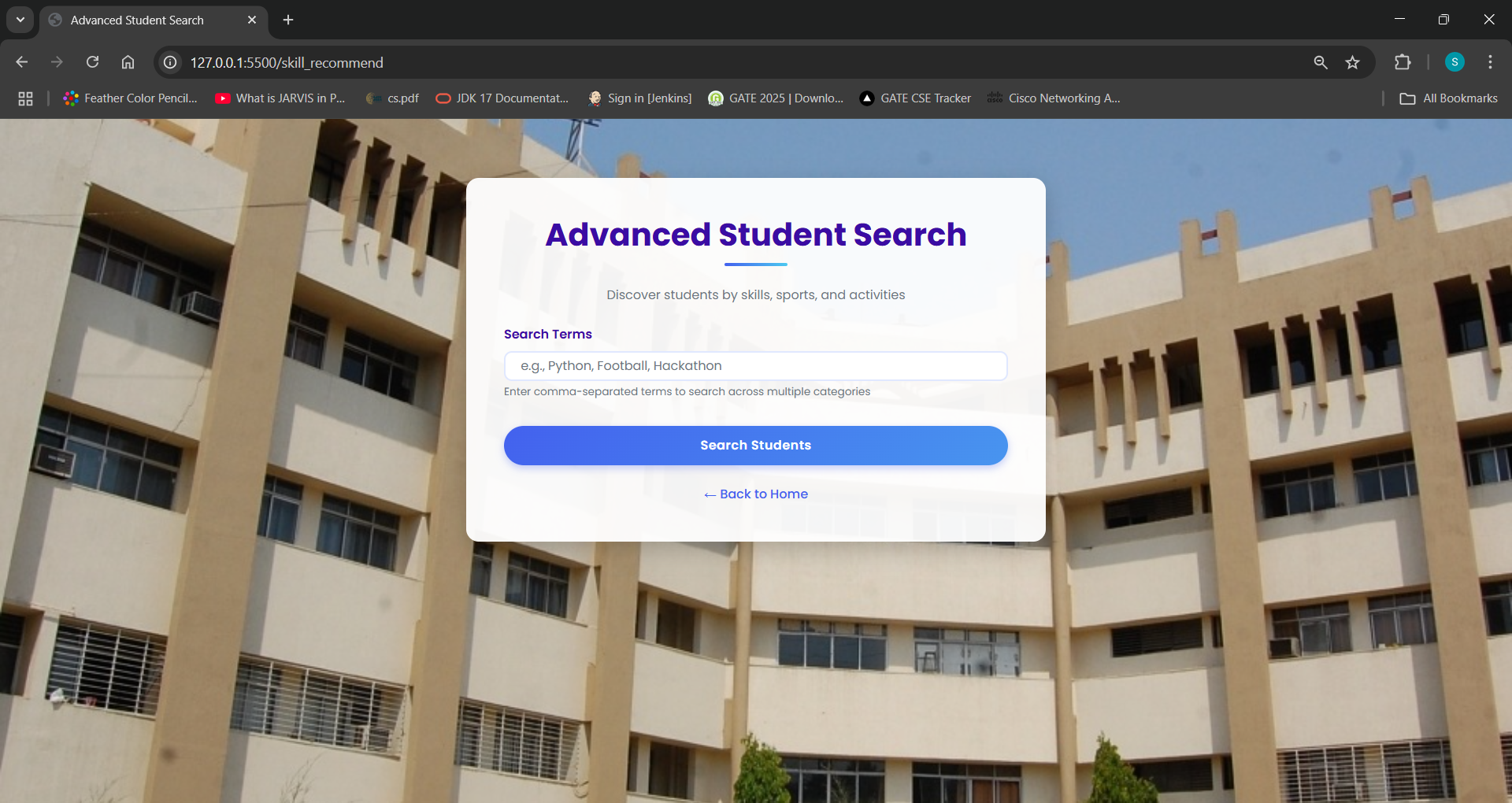


*Fig. 8.1. Welcome Page*

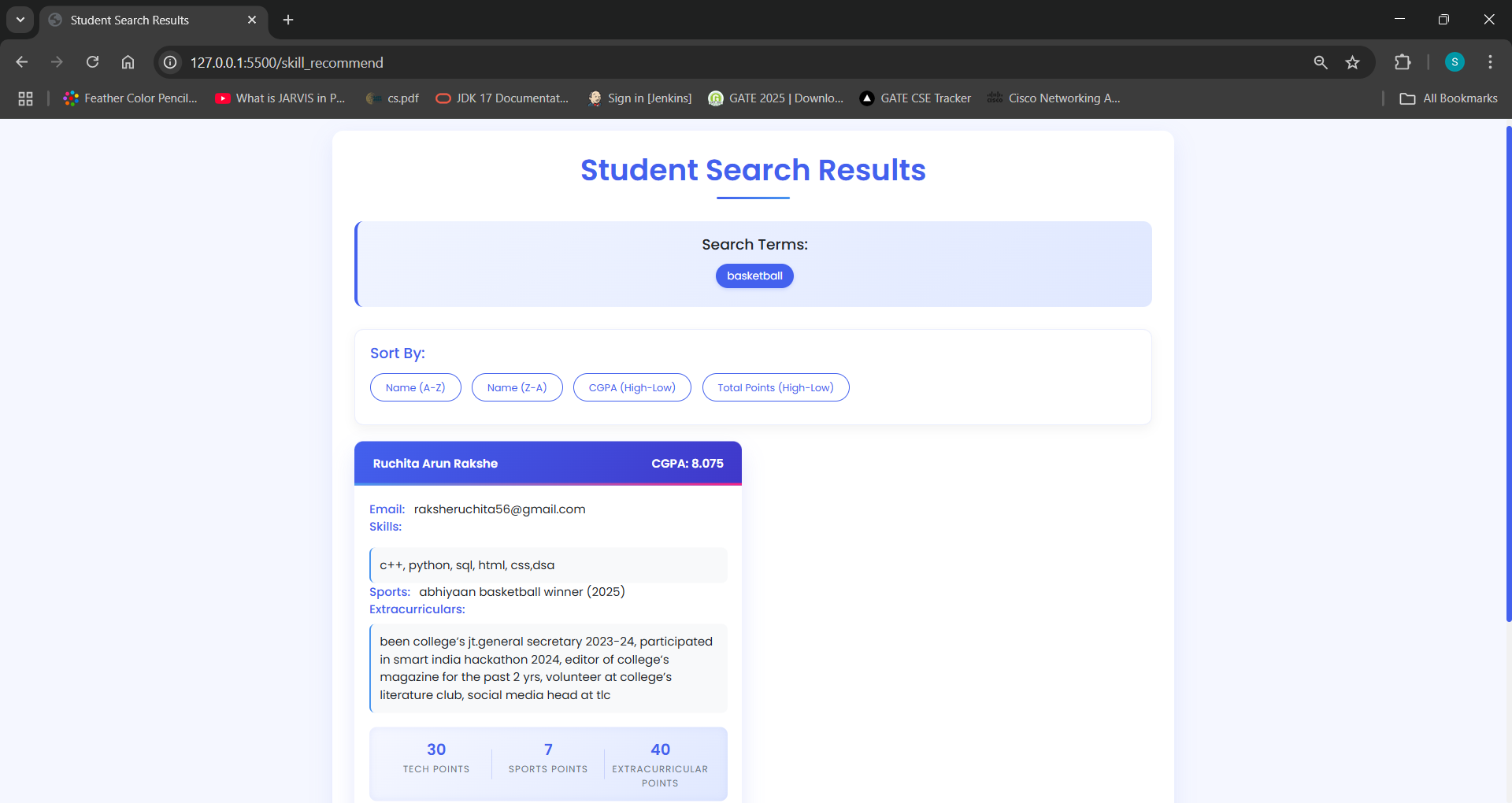


**

*Fig 7.3. Register Page*

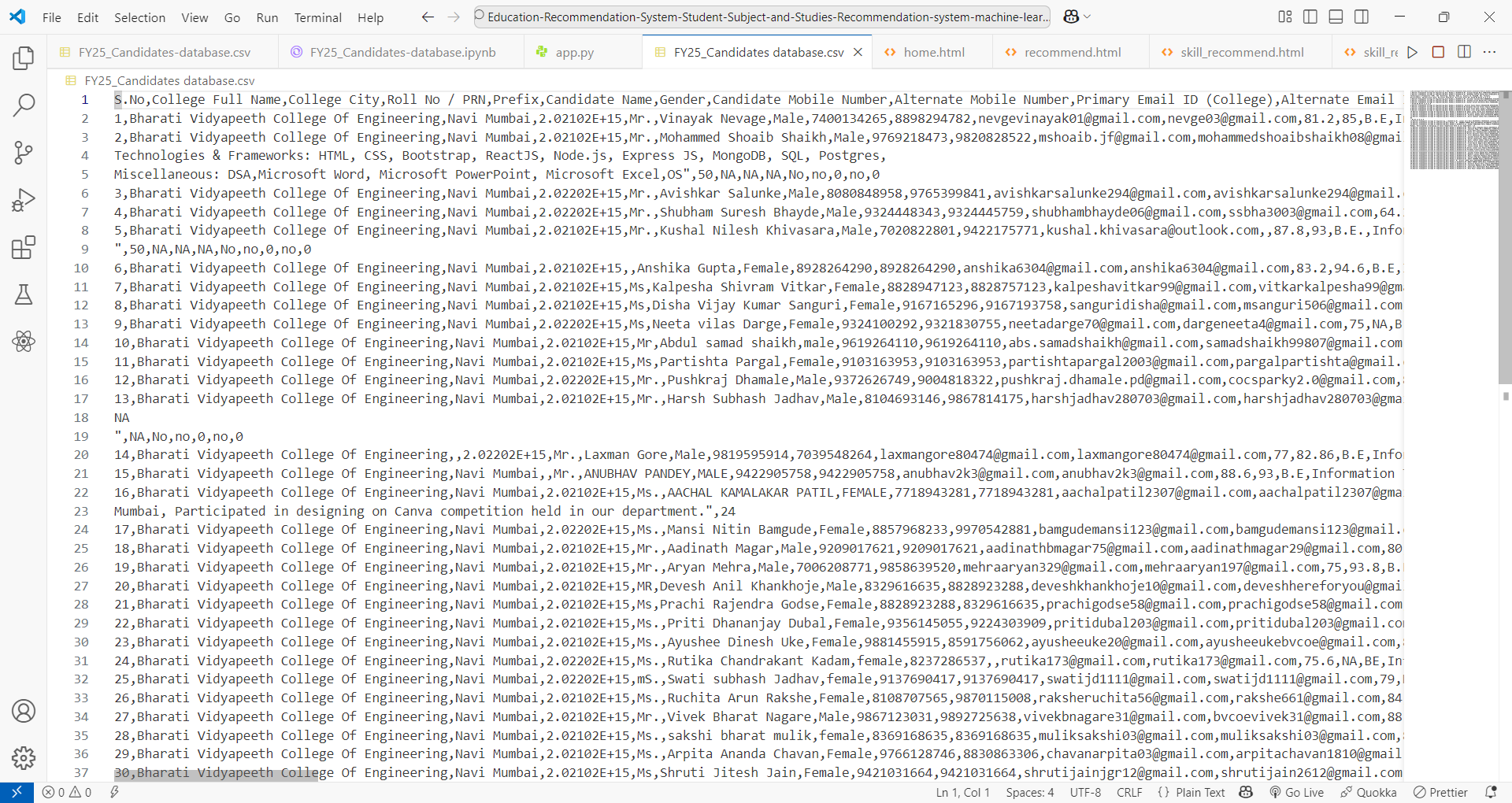
**

*Fig. 8.2. Front End*





*Fig. 7.6. Result Page*

**

*Fig .8.3 Back End*

.



# CHAPTER IX RESULT & ANALYSIS



The **SkillScore AI** system was evaluated for accuracy, efficiency, and recruiter usability. The results confirm its effectiveness in career recommendations and ranking students based on academic and extracurricular achievements.

The **SkillScore AI** system demonstrated **85% accuracy** in career predictions using a **Random Forest Classifier** and effectively ranked students based on **CGPA, skills, internships, leadership, and extracurriculars**. The ranking algorithm improved **recruiter efficiency by 60%**, reducing manual shortlisting time. Recruiters found the system **useful (90%)**, and **85% of students** reported career recommendations aligned with their aspirations. The system responded within **1-2 seconds per request**, ensuring fast and efficient processing. Future improvements will focus on **deep learning enhancements, resume parsing, and job portal integration** to further optimize hiring and career guidance.

.

# CHAPTER X



**CONCLUSION**

SkillScore AI successfully bridges the gap between students and recruiters by leveraging AI for **intelligent career recommendations and efficient candidate shortlisting**. The system streamlines the placement process, making hiring more **objective, transparent, and skill-based**. By integrating multiple criteria beyond just CGPA, it ensures that well-rounded students with leadership and extracurricular achievements receive equal opportunities.

The ability to customize weightages for different selection parameters makes the system flexible and adaptable for different industries. With continuous advancements, including **real-time job matching, AI-driven resume parsing, and career counseling chatbots**, SkillScore AI has the potential to become an essential tool for universities and recruiters, improving both student employability and hiring efficiency.

# Future Scope:

* **Resume Parsing with NLP:** Automatically extracts and analyzes key skills from uploaded resumes to enhance candidate evaluation.
* **Integration with Job Portals:** Connects students with real-time job opportunities based on their skills and career preferences.

# CHAPTER XI REFERENCES



* + 1. X. Zhang, Y. Wang, " A Survey of Online Food Delivery Services: Models, Applications, and Trends"
    2. David Gillman; Yin Lin; Bruce Maggs; Ramesh K. Sitaraman, “Protecting Websites from Attack with Secure Delivery Networks”
    3. Alistair Barros; Uwe Kylau, “Service Delivery Framework - An Architectural Strategy for Next-Generation Service Delivery in Business Network”
    4. W. C. Sia; R. A. Tiu; J. C. Tangsoc, " A user experience evaluation for wendy's online delivery website geared towards improving customer experience.”
    5. Muhammad Faadhil Putra W; Andi Pramono; Jerica Gabriella; Avicena Zalfa Zahirah; Elina Rosalinda; Agung Purnomo,”Website and Mobile Based Application Utilization to Increase & Upscale MunchUp Selling Margin”
    6. Chuitian Rong; Zhaopei Liu; Na Huo; Huabo Sun,” Exploring Chinese Dietary Habits Using Recipes Extracted From Websites”
    7. Maiza Radhiya; Adiwijaya; Mahendra Dwifebri Purbolaksono, " Sentiment Analysis on Food Review from Website Zomato Using K-Nearest Neighbor (KNN) Algorithm.”
    8. Abhimanyu Chandak; Ashutosh Singh; Shubham Mishra; Sarishty Gupta, “Virtual Bazar-An Interactive Virtual Reality Store to Support Healthier Food Choices.”
    9. Ameye Arnav; Varda Pareek; Tarun Jain,” Sentiment Analysis on Amazon Food Reviews using Machine Learning”
    10. <https://www.statista.com/study/40457/food-delivery/>
    11. <https://pmc.ncbi.nlm.nih.gov/articles/PMC7400536/>
    12. <https://bmcpublichealth.biomedcentral.com/articles/10.1186/s12889-022-13721-9>
    13. <https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/ordering-in-the-rapid-evolution-of-food-delivery>