**SUMMER TRAINING/INTERNSHIP**

**PROJECT REPORT**

(Term June-July 2025)

**Student Placement Predictor with Readiness Scoring System**

Submitted by

**Anmol Kannaujiya: 12314634**

**Nidhi Rana: 12309338**

**Sharmina P: 12323970**

**Shivadan Dogra: 12323642**

**Course Code : PETV79**

Under the Guidance of

**Assistant Professor Mahipal Singh Papola**

**School of Computer Science and Engineering**

**Bonafide Certificate**

Certified that this project report “**Student Placement Predictor with Readiness Scoring System**” is the Bonafide work of “Anmol Kannaujiya, Nidhi Rana, Sharmina P, Shivadan Dogra” who carried out project work under my supervision.

**Signature**

<Name of the Supervisor>

Anmol Kannaujiya

Nidhi Rana

Sharmina P

Shivadan Dogra

**Signature**

<<Signature of the head of the department>>

**Signature**

**<<Name>>**

**Head of department**

<<Signature of supervisor>>

**Acknowledgement:**

Our sincere appreciation goes out to Lovely Professional University School's School of Computer Science and Engineering for providing us, the undersigned team members, with the chance to take part in the Summer Industrial Training Program. This program has given us a great opportunity to improve our technical understanding, obtain practical experience, and apply what we've learnt in the classroom to real-world problems.

We are particularly appreciative of our trainer, Mahipal Singh Papola, for his knowledgeable direction, unwavering encouragement, and support during the training session. Their guidance was essential in assisting us in comprehending difficult ideas and finishing our job successfully.

We also want to express our gratitude to the entire professors and staff of the School of Computer Science for their unwavering support and for creating a stimulating environment for learning and creativity.   
We have improved our technical problem-solving skills, teamwork, and comprehension of machine learning applications as a result of this experience. We sincerely appreciate this chance.   
  
**Team Members:**

* Anmol Kannaujiya
* Nidhi Rana
* Sharmina P
* Shivadan Dogra

Table of content:

1. **Abstract**
2. **Introduction**
3. **Project Details**
   1. Project overview
   2. Problem definition
   3. Scope and objectives
   4. System Requirements
   5. Architecture Diagram (if any)
   6. Data flow / UML Diagrams
4. **Implementation**

4.1 Tools used

* 1. Methodology
  2. Modules / Screenshots
  3. Code snippets (if needed)

1. **Results and Discussion** 
   1. Output / Report
   2. Challenges faced
   3. Learnings
2. **Conclusion**
   1. Summary
3. **Abstract:**

The student placement prediction system aims to estimate the likelihood of a student securing a job through campus recruitment by analyzing a combination of academic, technical, and behavioral factors. In today’s competitive educational environment, relying solely on traditional indicators like CGPA or aptitude scores is no longer sufficient. To bridge this gap and better support both students and academic institutions, it becomes essential to leverage data-driven approaches that provide deeper, actionable insights into employability.

This project analyzes a structured student dataset containing key parameters such as CGPA, internship experience, number of projects, workshop participation, aptitude scores, soft skill ratings, extracurricular involvement, and training attendance. The goal is to develop and fine-tune a machine learning model capable of accurately forecasting placement outcomes based on these inputs. Classification algorithms including Logistic Regression, Random Forest, and XGBoost were trained and evaluated to determine the best-performing model.

The final model is deployed through a user-friendly Streamlit web application, which enables real-time placement predictions based on student inputs. Additionally, the system generates a “Placement Readiness Score” ranging from 0 to 100, providing personalized suggestions to improve individual student profiles. By combining machine learning with an interactive user interface, this project empowers students and institutions to make more informed, data-backed placement decisions.

.

1. **Introduction:**

Campus placements are a significant turning point for students in today's highly competitive academic and professional environment, influencing their move from academia to the workplace. However, as the number of graduates rises year and recruiters' expectations rise, it is critical that both students and institutions evaluate placement preparedness fairly. Conventional measures like CGPA or aptitude scores are frequently simplified to assessing employable talents and traits. Therefore, a data-manual, intelligent strategy is needed to analyse and anticipate placement results.

This project under the title of student placement prophet, an introduction of a machine learning -based solution that it uses supervised classification algorithm to guess whether a student can be placed based on a combination of educational, technical and behavioural characteristics. Datasets used to train the model include characteristics such as CGPA, internship experience, project count, workshop participation, aptitude test score, soft skill rating, extra curricular arrival, and training participation. After preparing the data and encoding the range of classes, many algorithms were trained and compared - namely to identify the most accurate models for logistics region, random forest, and xgboost -prediction.

Once after selecting and evaluating the optimal model, an interactive web application was developed using a streamlight. This interface allows users to input students data and get immediate predictions on placement possibilities. In addition to the binary classification (placed or not), the system produces a placement readiness score (0–100), which reflects the overall profile power of the student. Also, app offers personalized improvement suggestions based on weaknesses identified in the input.

The primary dataset used in this project is a structured student placement dataset that includes educational and experiential characteristics from hundreds of students with various educational backgrounds. Each record represents a student profile and includes CGPA, number of internships, number of educational projects, aptitude test score, soft skill rating, extra-curricular activity participation, placement training appearance, and secondary and higher secondary educational marks (SSC and HSC). This dataset provides a well-balanced mixture of numerical and classified data, which enables the training of accurate and explanatory machine learning models. Educational placement reports and research papers were prepared to identify relevant predictions and industry-handled employment factors.

1. **Project Detail:**

3.1 Project overview:

Aviation industry faces the challenge of ensuring flight safety when working in an environment shared with wildlife, especially birds. Wildlife attacks - collisions between merits and animals, mainly birds - use a significant protection and financial risk. In response to this challenge, the purpose of our project is to develop a forecast machine learning model that estimates the economic impact of the plane wildlife attacks, and provides interactive visualization that helps identify high -risk airports and time.

Our "Artificial Intelligence and Machine Learning" curriculum incorporates the project. This is the process of using theoretical ideas to tackle a practical aviation security challenge. The necessity to use AI for ecological danger assessment, operational effectiveness, and safeguarding the public is what drives us.   
The FAA (Federal Aviation Administration) reports that over 14,000 bird assaults occur in the United States each year, causing aircraft damage, emergency landings, delays, and expensive repairs. In a well-known instance, American Airways Flight 1549's emergency landing on the Hudson River was caused by a bird strike.   
Our objective is to address this urgent problem by: utilizing machine learning to forecast the financial cost of a wildlife strike.

* determining trends in the frequency of strikes according to weather, species, location, time, and aircraft type.
* delivering useful insights to aviation authorities via data-driven visualizations.

**3.2 Problem Definition:**

Getting hired through university recruitment is a big step for a student's academic career. Nonetheless, the procedure is really competitive and impacted by a number of academic, technical, and personal growth elements. The employability of a student cannot be fully determined by looking only at their CGPA or aptitude results. Therefore, the goal of this project is to develop a machine learning-based predictive system that uses a comprehensive profile—which includes academic accomplishments, soft skills, certifications, internships, and extracurricular activities—to assess a student's placement chances.

The "Artificial Intelligence and Machine Learning" course included this project, which uses theoretical ideas like learning under supervision, data preprocessing, model modification, and performance evaluation in a practical, educational context. By offering practical insights into their placement preparedness and pinpointing areas for development, the main objective is to empower students..

To achieve this, a classification-based prediction model was developed using input features like CGPA, internship count, project involvement, aptitude test scores, soft skill ratings, and participation in training or extracurricular activities. The model was deployed as an interactive web application using Streamlit. It not only predicts whether a student is likely to be placed but also generates a Placement Readiness Score (ranging from 0–100), which reflects the strengths and gaps in a student’s profile, along with targeted suggestions.

Our project workflow consists of the following steps:

1. Data Collection: FAA dataset in CSV format.
2. This phase included removing invalid or inconsistent student records, handling missing values, and encoding the range (e.g., yes/no response to additional activities and training participation). The feature scaling was applied to ensure uniformity in features such as numerical inputs such as CGPA, Aptitude score and educational digits.

3. Searching data analysis (EDA):

The EDA was operated to understand the relationship between various characteristics and placement results. This included identifying trends among non-plated students placed on the basis of CGPA, internship and other factors. Visualizations were created to highlight the major insight such as soft skills, aptitude score and the impact of training appearance on placement success.

4. Model Building:

A classification model was designed to guess whether a student would be kept. Several machine learning algorithms were tested, including logistics Regression, Random Forest Classifier and XGBOOST classifier. The feature importance was also analyzed to determine which properties the placements affect the results the most strongly.

5. Model Evaluation:

The model was evaluated using accuracy score, accurate, recall, F1-score and confusion matrix. In models, Xgboost provided the best performance with more than 78%accuracy, which gained a strong future power and balanced classification in the categories of and non-tactics.

6. Deployment Ready Model:

The final selected model was integrated into a user -friendly web application using stream light. The app accepts the user input through a form, predicts the placement possibility, calculates a placement readiness score (0–100), and provides.

The development of this project was carried out using Python, a versatile and widely adopted programming language in the field of data science and artificial intelligence. For efficient data handling and preprocessing, we used pandas and NumPy to manage structured data, handle missing values, and perform numerical transformations. Exploratory Data Analysis (EDA) and feature interpretation were performed using visualization libraries such as matplotlib and seaborn, which enabled the identification of patterns and correlations within the dataset.

The machine learning models including Logistic Regression, Random Forest Classifier, and XGBoost — were built using the scikit-learn and XGBoost libraries. These models were trained to classify students based on their placement outcomes. Model evaluation was carried out using classification metrics such as accuracy, precision, recall, F1-score, and confusion matrix to ensure a balanced and reliable prediction across both classes.

The trained model was serialized using pickle and integrated into a web application using Streamlit, an open-source Python framework for building intuitive and interactive user interfaces. The entire workflow, from data preprocessing to deployment, was implemented in Jupyter Notebook, which provided an interactive environment for developing, testing, and visualizing results in real-time.

. The goal of this project is to develop a machine learning based predictive model that can assess a student's likelihood of being placed in a campus recruitment drive based on multiple academic, experiential, and behavioural attributes. Traditional evaluation methods often rely solely on academic scores, overlooking the combined impact of other important factors such as soft skills, internships, certifications, and aptitude.

By analysing historical student data including CGPA, internship experience, project count, aptitude test score, soft skills rating, academic marks, and participation in extracurriculars and training the model aims to predict whether a student is likely to be placed. In addition to binary prediction, the system also provides a Placement Readiness Score (ranging from 0 to 100), which reflects the student’s overall employability profile.

This project also seeks to provide actionable insights into which factors most significantly impact placement outcomes and how students can enhance their chances of getting placed. The use of data-driven visualization and interactive prediction tools can help students self-evaluate and prepare more effectively, while also aiding academic institutions in identifying readiness gaps across the student body.

Campus placements play a critical role in shaping students’ future careers, yet predicting who gets placed and why remains uncertain. Many students face challenges despite having acceptable academic performance, while others succeed due to a stronger combination of soft skills, training, and experience. This project addresses a growing need in education and career readiness by applying machine learning to help students understand and improve their placement outcomes. Key reasons to solve this problem include:

- Predictability: Placement outcomes depend on structured, measurable attributes such as CGPA, projects, aptitude, and internships making this an ideal use case for machine learning.

- Skill Gap Identification: Students may not realize which parts of their profile need improvement. An intelligent system can provide targeted, personalized feedback.

-  Career Enablement: Predictive insights and readiness scoring can help students prepare more effectively, boosting confidence and performance during campus recruitment drives.

-  Institutional Support: Academic advisors and placement cells can use such a system to identify trends and provide early guidance for at-risk students.

**Challenges in the Raw Problem:**

Several factors make the student placement prediction problem both interesting and non-trivial:

-  Class Imbalance: The number of students placed is often significantly different from those not placed, which can bias classification models toward the majority class.

-  Categorical Data: Features like training participation, extracurricular involvement, and certification status are categorical and require proper encoding for model compatibility.

-  Missing or Incomplete Records: Real-world educational datasets may include missing entries or inconsistent responses for fields like soft skills ratings or project experience.

-  Interpretability: Since the results are used to guide student development, the model must not only be accurate but also offer transparent insights into which features impact predictions.

We approached this problem as a supervised classification task, where the input features (X) include both numerical (e.g., CGPA, aptitude score, SSC and HSC marks) and categorical variables (e.g., training, extracurriculars), while the target variable (y) is the placement status (Placed or Not Placed). Additionally, a secondary Placement Readiness Score (0–100) was developed to reflect the strength of a student's overall profile in a quantifiable manner.

The project also integrates exploratory data analysis (EDA) and visualizations to explore trends, feature correlations, and performance metrics , helping both students and academic advisors understand what drives successful placements.

* 1. Project Scope and Objectives

This project operates at the intersection of education, data science, and interactive AI applications. Its scope extends from understanding and preprocessing real-world student academic data to building a machine learning model for placement prediction and finally deploying this model through a Streamlit web application for end-user interaction.  
The core dataset sourced from structured student records contains numerous entries spanning various educational profiles and includes diverse variables such as CGPA, number of internships, projects, aptitude scores, soft skills, extracurricular activities, and placement training participation. Given its complexity and real-world nature, this dataset provides an excellent opportunity to apply AI/ML for predictive insights with practical impact.  
The project is designed to support:  
• Students aiming to evaluate and improve their placement readiness  
• Placement coordinators and training & development teams for early intervention  
• Academic counsellors seeking data-driven student insights  
• Educational administrators aiming to improve institutional placement performance  
Moreover, the project demonstrates a complete machine learning lifecycle, from raw data to a functioning application available for public or institutional use. The inclusion of the Streamlit app bridges the gap between model development and real-world usability.

Core Objectives:

The project is guided by a set of well-defined objectives to ensure both technical and practical relevance:

1. Data Understanding and Preprocessing involving importing , cleaning , and preprocessing the student placement dataset, and also handling missing values, encoding categorical data (e.g., extracurriculars, training), scaling numerical features, and preparing the dataset for model input.
2. Exploratory Data Analysis (EDA) involving visualization of placement trends based on CGPA, internships, aptitude scores, projects, and soft skills. Identifying key features that influence placement outcomes and examining differences between placed and non-placed students.
3. Model Development: Train and evaluate classification algorithms such as Logistic Regression, Random Forest Classifier, and XGBoost to predict placement outcome. Optimize for accuracy using metrics like Accuracy Score, F1-Score, and Confusion Matrix. Save the trained model using pickle for future reuse and deployment.
4. Web Application Deployment with Streamlit. To build a user-friendly interface using Streamlit to allow users to input student profile details (e.g., CGPA, aptitude score, internships, projects). View the placement prediction and readiness score in real-time using the trained model. Include intelligent input handling: sliders, number inputs, select boxes, and dropdowns. Ensure the input is transformed to match model training format and display output in a clear and meaningful format.
5. Practical Integration and Accessibility: Package the solution for deployment to Streamlit Cloud or local academic servers. Make the tool accessible to non-technical users, especially students and training coordinators. Allow for potential future integration with institutional databases or student portals.

Included in scope:  
The scope of this project includes data cleaning, feature engineering, and the development of classification models to predict student placement outcomes. It also involves building a Streamlit web application that allows users to input student information and receive real-time predictions and readiness scores. The app handles data preprocessing, feature transformation, and model loading using pickle, making it   
a complete end-to-end solution deployable on platforms like Streamlit Cloud.

* 1. System requirements:

Hardware requirements: To run both the machine learning pipeline and the Streamlit web application efficiently, the following hardware specifications are recommended:

* Processor: Intel Core i5 or higher / AMD Ryzen 5 or higher
* RAM: Minimum 8 GB (16 GB recommended for faster training and smooth multitasking)

• Storage conditions: To hold datasets, model records, and records, there must be a minimum of 2 gigabytes of free disc space.   
• Display: any device with a standard resolution (HD or greater is advised for an improved user interface).   
• Internet connection: necessary for online Streamlit app deployment or external API access (optional)   
Software prerequisites: Streamlit, Python, and a number of data science libraries were used to create this project. The software and package requirements are listed below:

**Operating System**

* Windows 10/11, Linux, or macOS (Python 3.8+ compatible)

**Python Environment**

* Python version: **3.8 or above**

Required Python libraries:

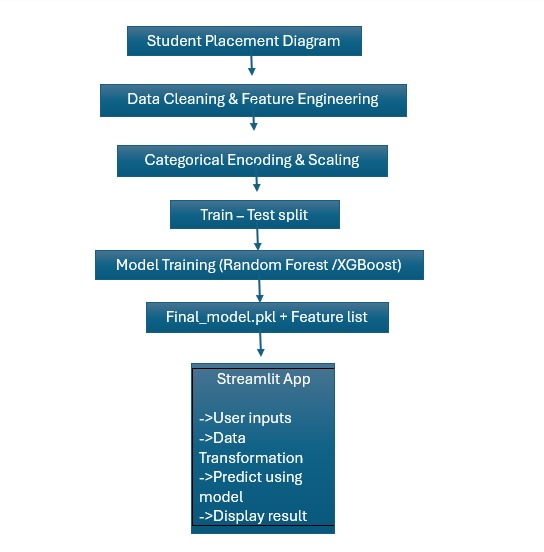
| **Library** | **Purpose** |
| --- | --- |
| pandas | Data loading, cleaning, and manipulation |
| NumPy | Numerical operations |
| scikit-learn | Model training and evaluation |
| matplotlib | Data visualization (optional in backend) |
| seaborn | Advanced plots (optional for EDA) |
| joblib | Model saving and loading |
| streamlit | Web application interface |
|  |  |

|  |  |
| --- | --- |
| To install all dependencies command required:  pip install -r requirements.txt |  |

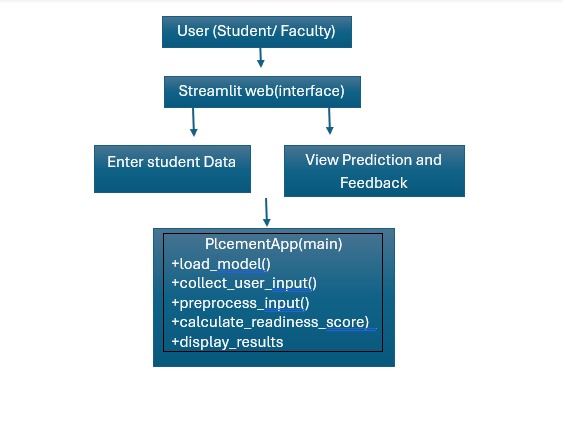
**Hosting and Deployment Requirements:**

To deploy the model using **Streamlit**, you can either host it locally or on a cloud platform like **Streamlit Cloud**. The only requirement is to ensure that:

* The final pkl file is placed correctly in the working directory
* All required columns (features used in training) are present in the prediction pipeline
* The Streamlit script (app1.py) runs without errors via: streamlit run app1.py
  1. Architecture Diagram:



* 1. UML



**4 Implementation**

4.1 Tools Implemented:

With the goal to facilitate the end-to-end pipeline of analysis of data, model creation, and real-time prediction via a web interface, the project was constructed utilising an assortment of crucial Python libraries and deployment tools.

Python was initially chosen as the primary programming language because of its broad support for activities involving data science and machine learning. Pandas and NumPy served as tools for preprocessing and data management. The CSV dataset was read, and data was explored using.head(),.info(),.shape, and.describe(), and cleaning procedures were carried out using.replace(),.dropna(), and.fillna().

For visualization during EDA, although not included in the final app, Matplotlib and Seaborn were utilized to analyze data distributions, feature correlations, and placement trends. The machine learning models were built using Scikit-learn and XGBoost, with Random Forest and XGBoost Classifiers chosen for their accuracy and reliability on structured student profile data. Model performance was assessed using Accuracy, Precision, Recall, F1-Score, and the Confusion Matrix. The trained model was serialized using Pickle and saved as final\_model.pkl.

To make the model accessible, we used Streamlit, a lightweight web framework that allows data science applications to be turned into interactive web apps with minimal effort. The file app.py uses Streamlit widgets such as number\_input, selectbox, and slider to collect student profile inputs and display prediction results. The user input is processed, transformed to match the model’s expected format, and passed to the model to return a placement prediction and readiness score in real time.  
Finally, development and experimentation were carried out in Jupyter Notebook, which supported interactive coding, visualization, and iterative model building in a structured and readable format.5.2 Methodology

The methodology followed in this project reflects a complete AI/ML development pipeline, moving systematically from data collection and preprocessing to model training, evaluation, and deployment through a user interface. Each phase was carefully designed to ensure data integrity, model accuracy, and usability.

1. Data Collection and Exploration  
The dataset used is a structured CSV file (placementdata.csv) containing student placement records with features such as CGPA, SSC/HSC marks, internships, projects, workshops, aptitude scores, soft skills, training participation, and extracurriculars.  
Exploratory Data Analysis (EDA) was performed using commands like .head(), .info(), and .describe() to inspect the structure, identify missing values, and understand distributions.  
Basic cleaning involved:  
df.dropna(), df.duplicated(), df.drop(['unnecessary\_column'], axis=1)

2. Data Cleaning and Preprocessing  
Key steps included:  
• Replacing missing values using .fillna()  
• Converting 'Yes'/'No' fields using .replace()  
• Encoding categorical variables via LabelEncoder/OneHotEncoder  
• Filtering invalid entries and ensuring complete data for all records

3. Model Building and Evaluation  
Supervised classification models were trained to predict placement outcomes. Models tested:  
• Logistic Regression  
• Random Forest Classifier  
• XGBoost Classifier  
The dataset was split into train-test sets using train\_test\_split(). Models were trained (.fit()) and tested (.predict()) with performance evaluated using accuracy, precision, recall, F1-score, and confusion matrix.  
XGBoost showed the highest accuracy (~78%) and was selected for deployment.

4. Model Deployment and Streamlit Interface  
The final model was saved using Pickle and integrated into a web interface using Streamlit.  
Workflow:  
• User inputs captured via Streamlit widgets (st.number\_input, st.selectbox)  
• Inputs processed to match trained model format  
• Prediction generated using:  
 with open('final\_model.pkl', 'rb') as f:  
  model = pickle.load(f)  
 prediction = model.predict(input\_array)  
• Outputs include placement result, readiness score (0–100), and personalized feedback

This app enables students and educators to access real-time placement predictions in an intuitive, browser-based interface.

4.2 Methodology:

The project follows a structured machine learning pipeline starting with data collection, preprocessing, model training, and deployment. The dataset, stored in CSV format, contains features like CGPA, internships, projects, aptitude scores, soft skills, and academic marks.

Preprocessing included handling missing values, encoding categorical variables (e.g., training, extracurriculars), and normalizing numerical features. Feature selection and cleaning ensured data quality and model readiness.

Multiple classification models were trained and evaluated: Logistic Regression, Random Forest, and XGBoost. The dataset was split into training and testing sets, and model performance was assessed using accuracy, precision, recall, and F1-score.

The best-performing model was saved using Pickle and deployed using Streamlit. The app accepts student inputs, performs real-time predictions, and provides a Placement Readiness Score along with personalized suggestions to guide improvement.

4.3 Screenshots:

- EDA:

A screenshot of a computer

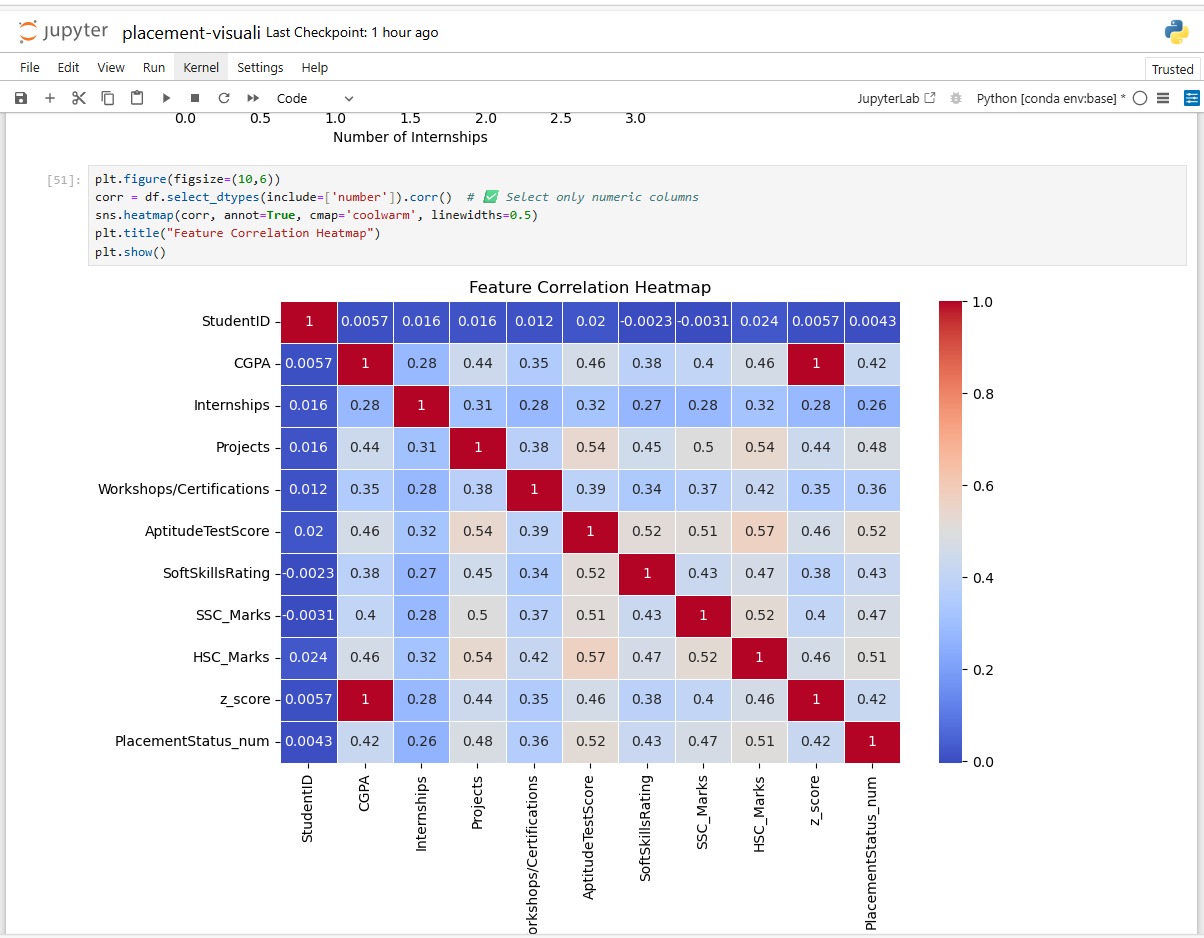
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



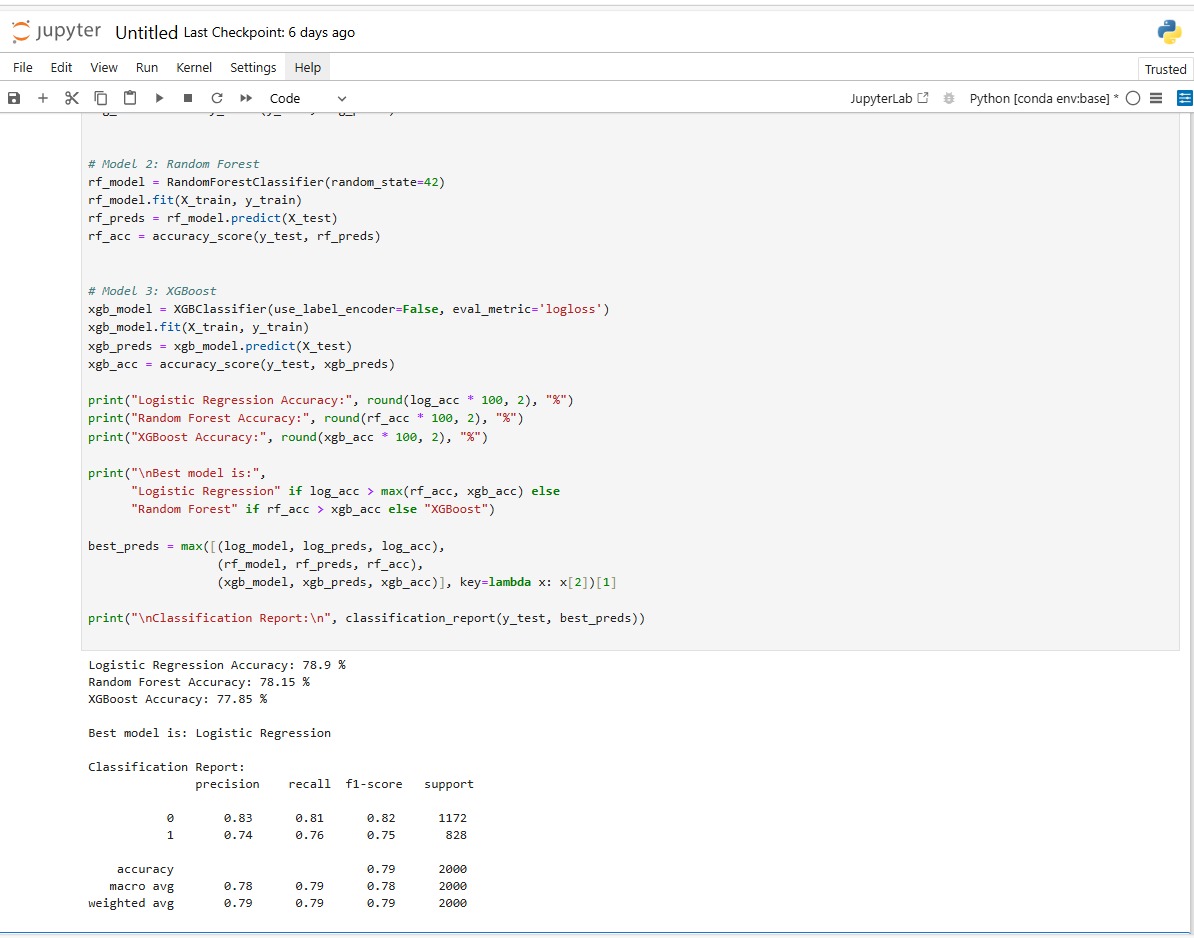
A screenshot of a computer

AI-generated content may be incorrect.

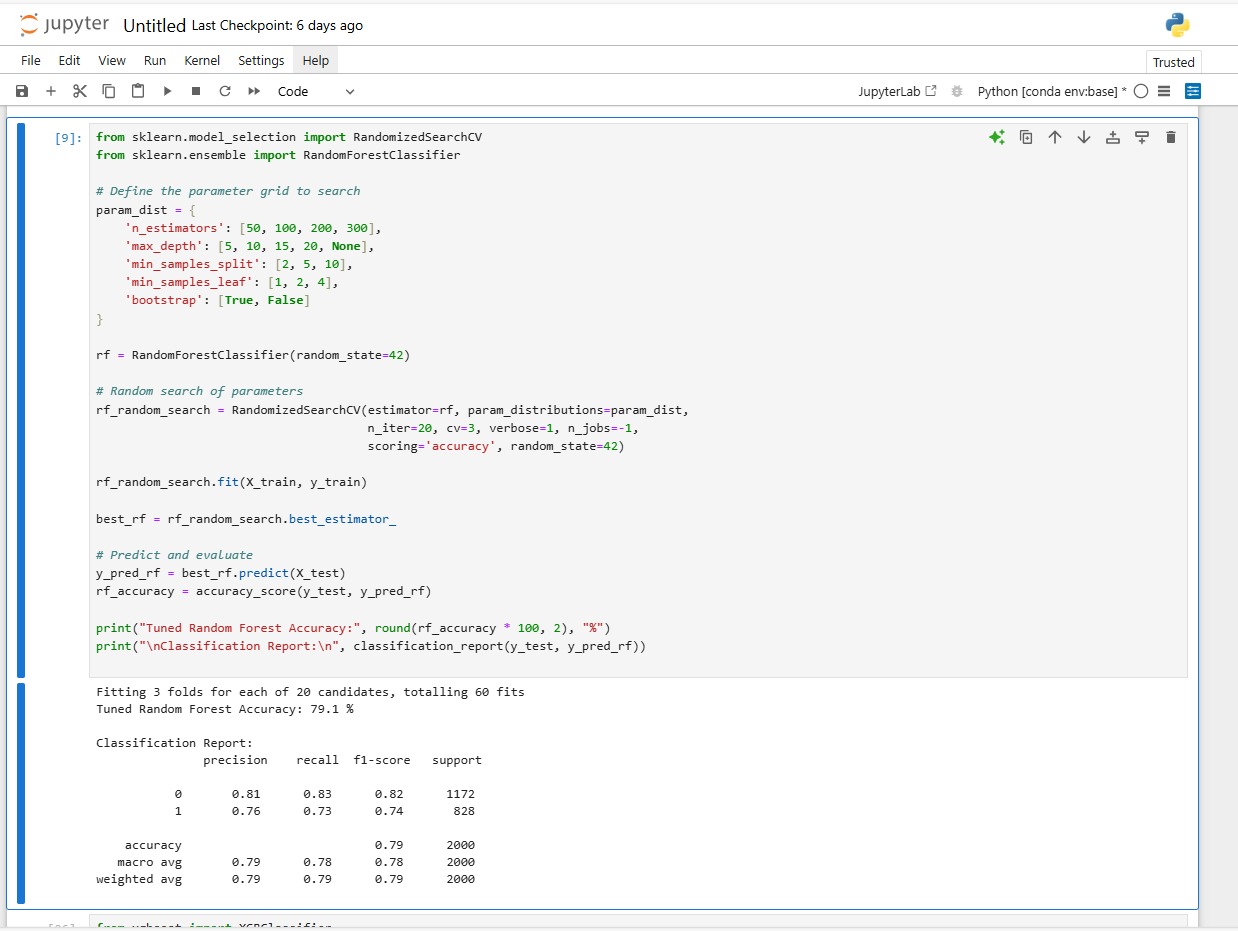
A screenshot of a computer

AI-generated content may be incorrect.

* + MODEL BUILDING(Without Hyperparameter Tuning) :



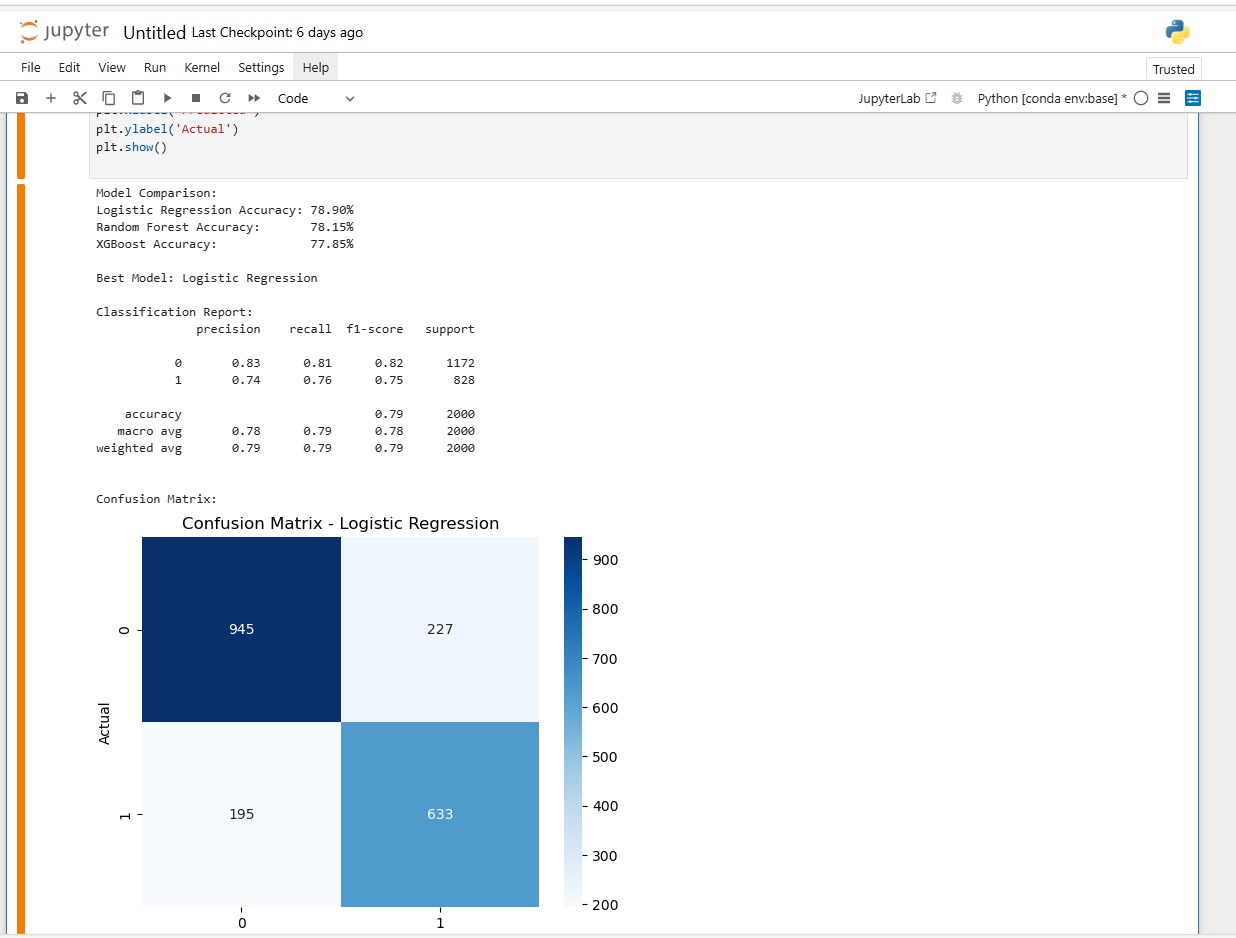
* + HYPRERPARAMETER TUNING:

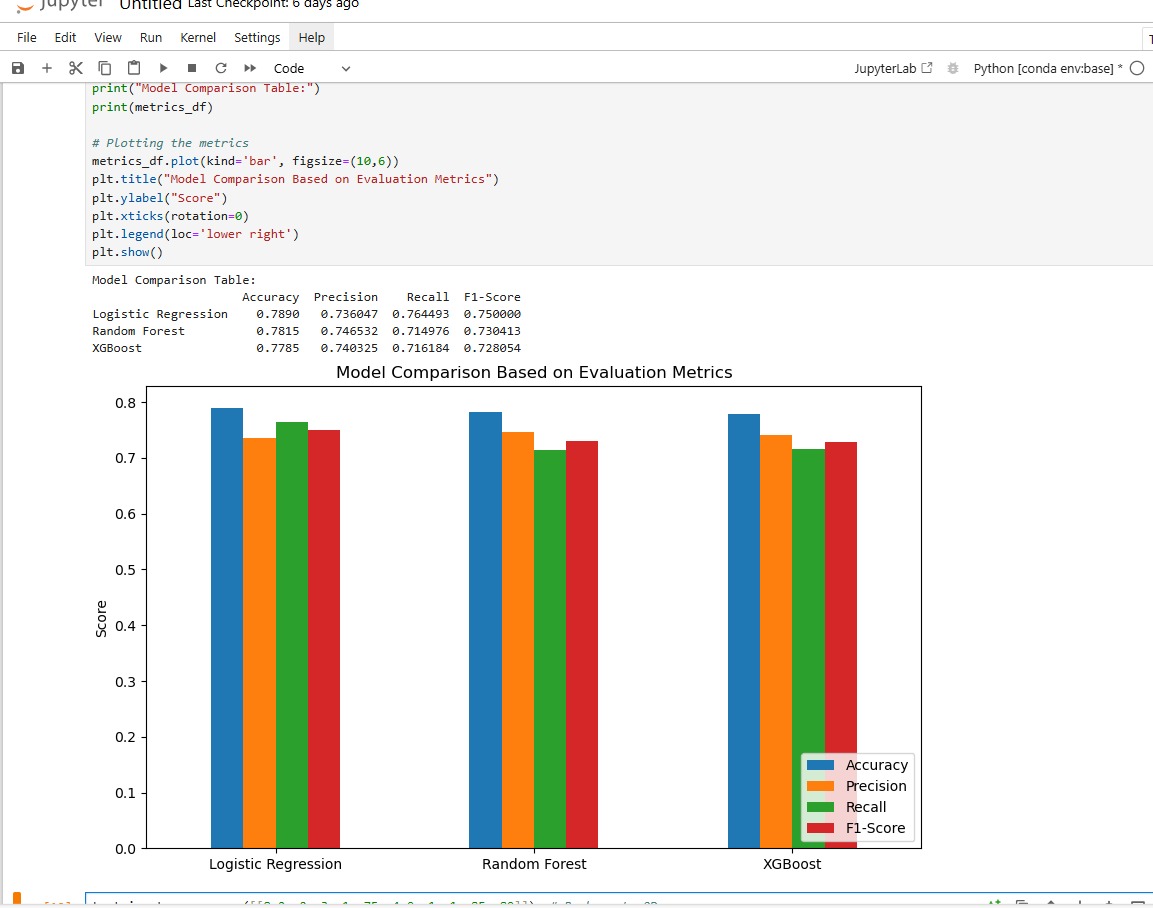


A screenshot of a computer

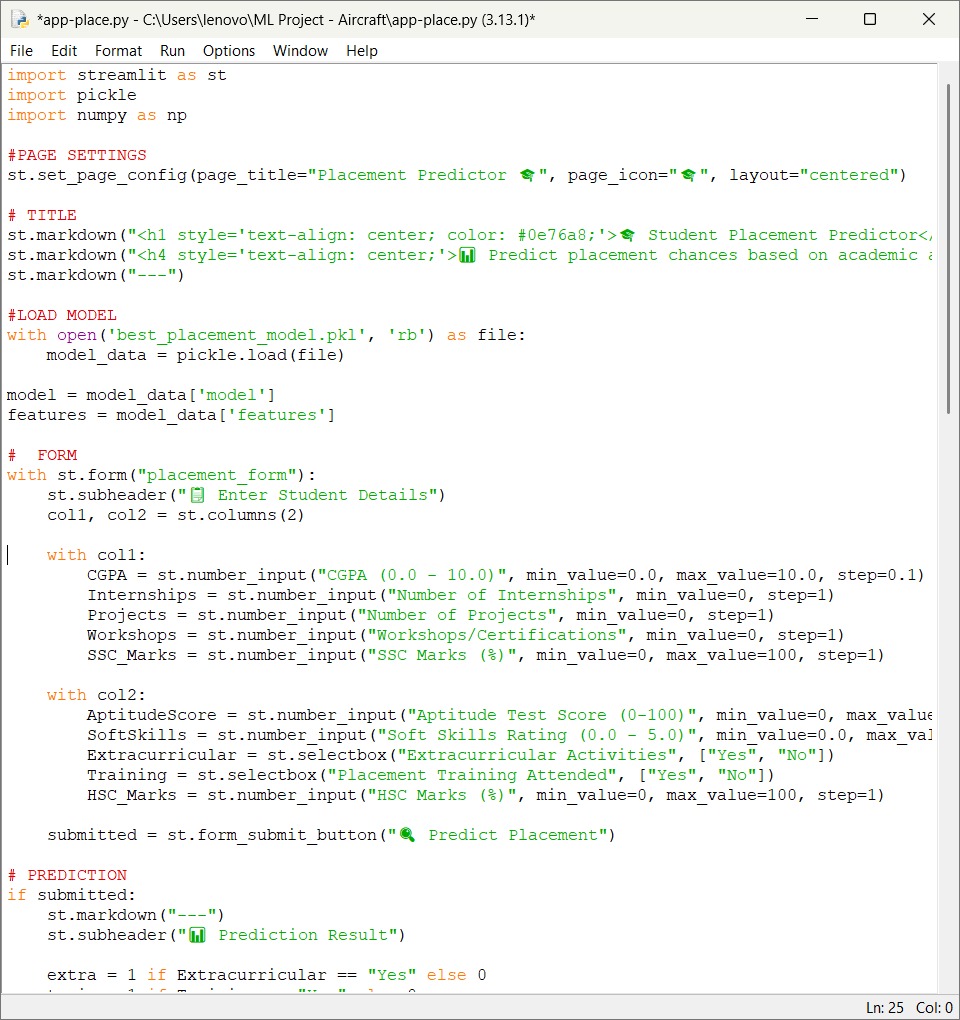
AI-generated content may be incorrect.

* + MODEL COMPARISON:



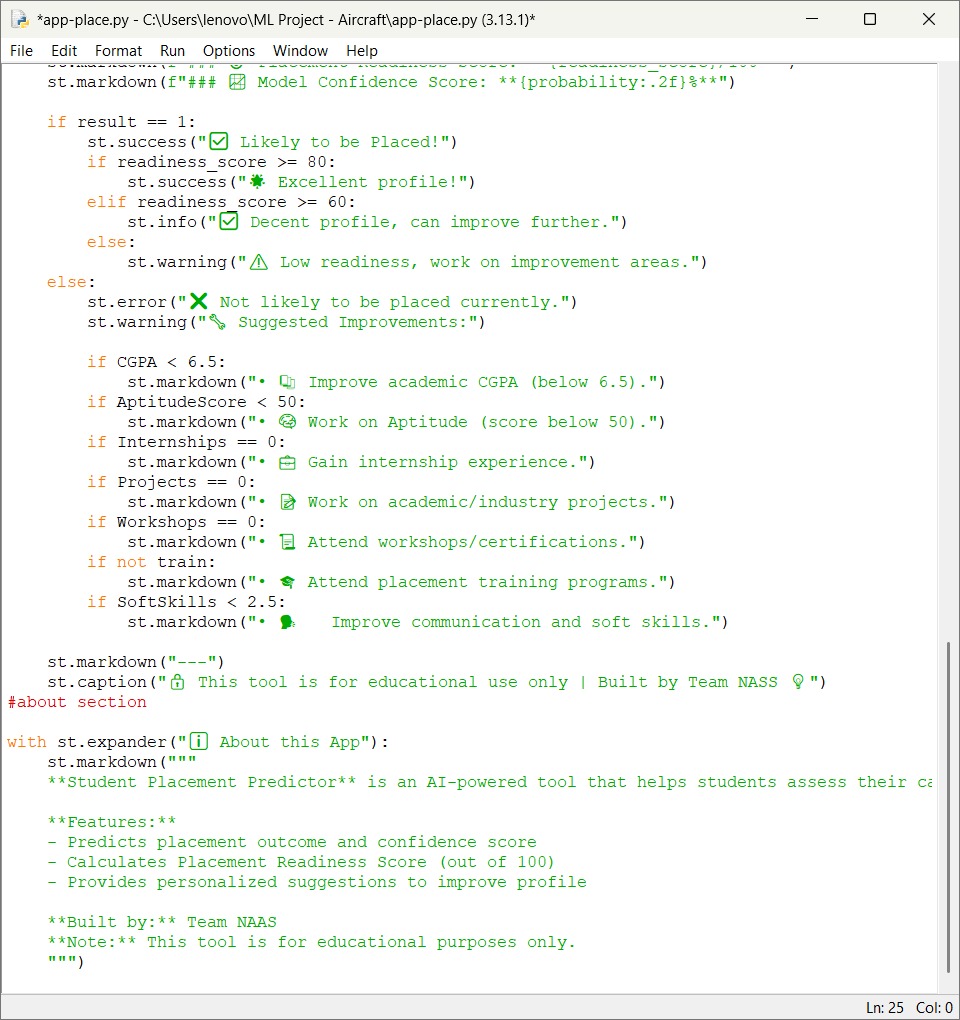


* + DEPLOYMENT (Streamlit):



**5 Results and Discussion**  
5.1 Output / Report

(Deployment )

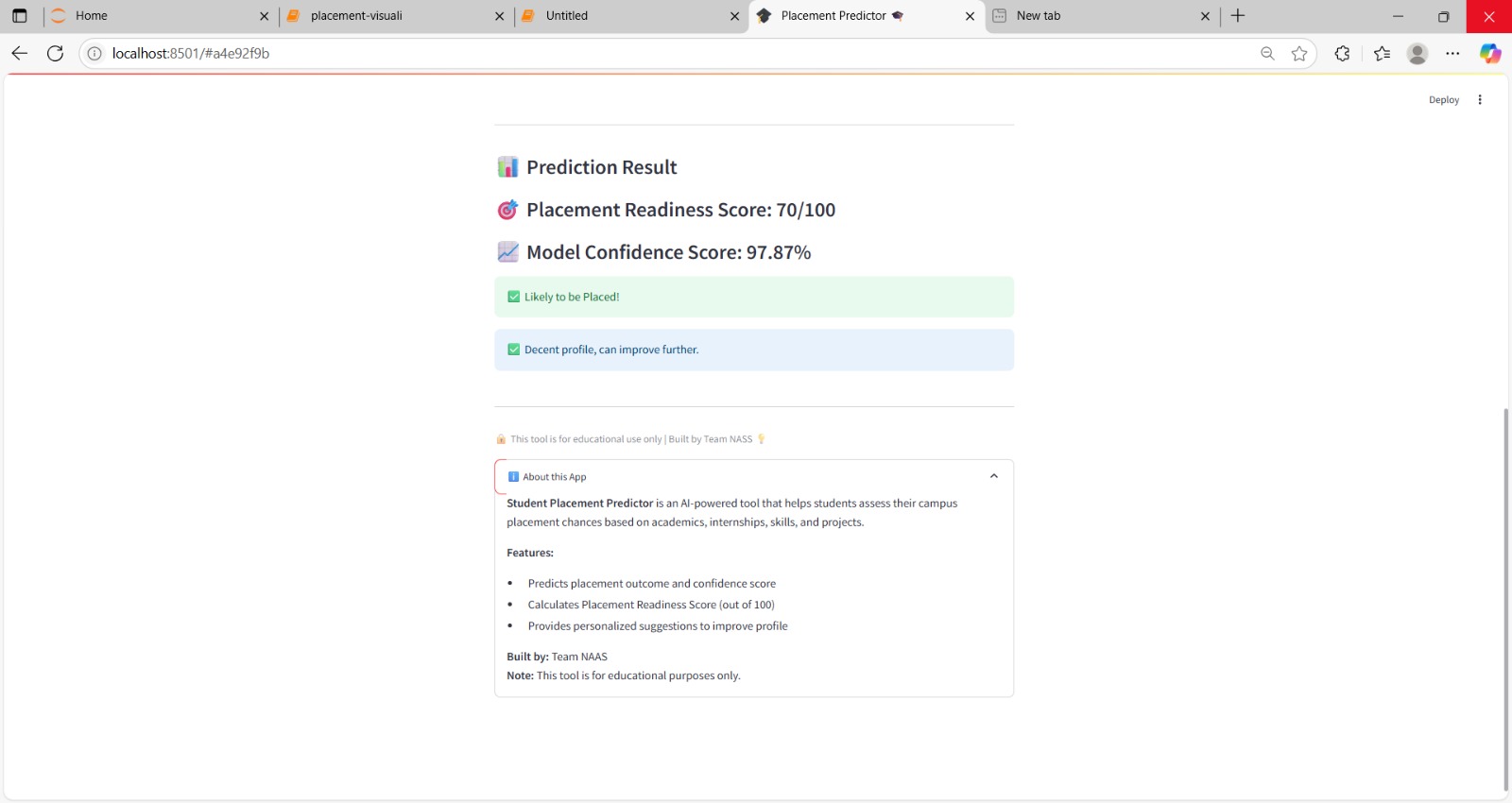


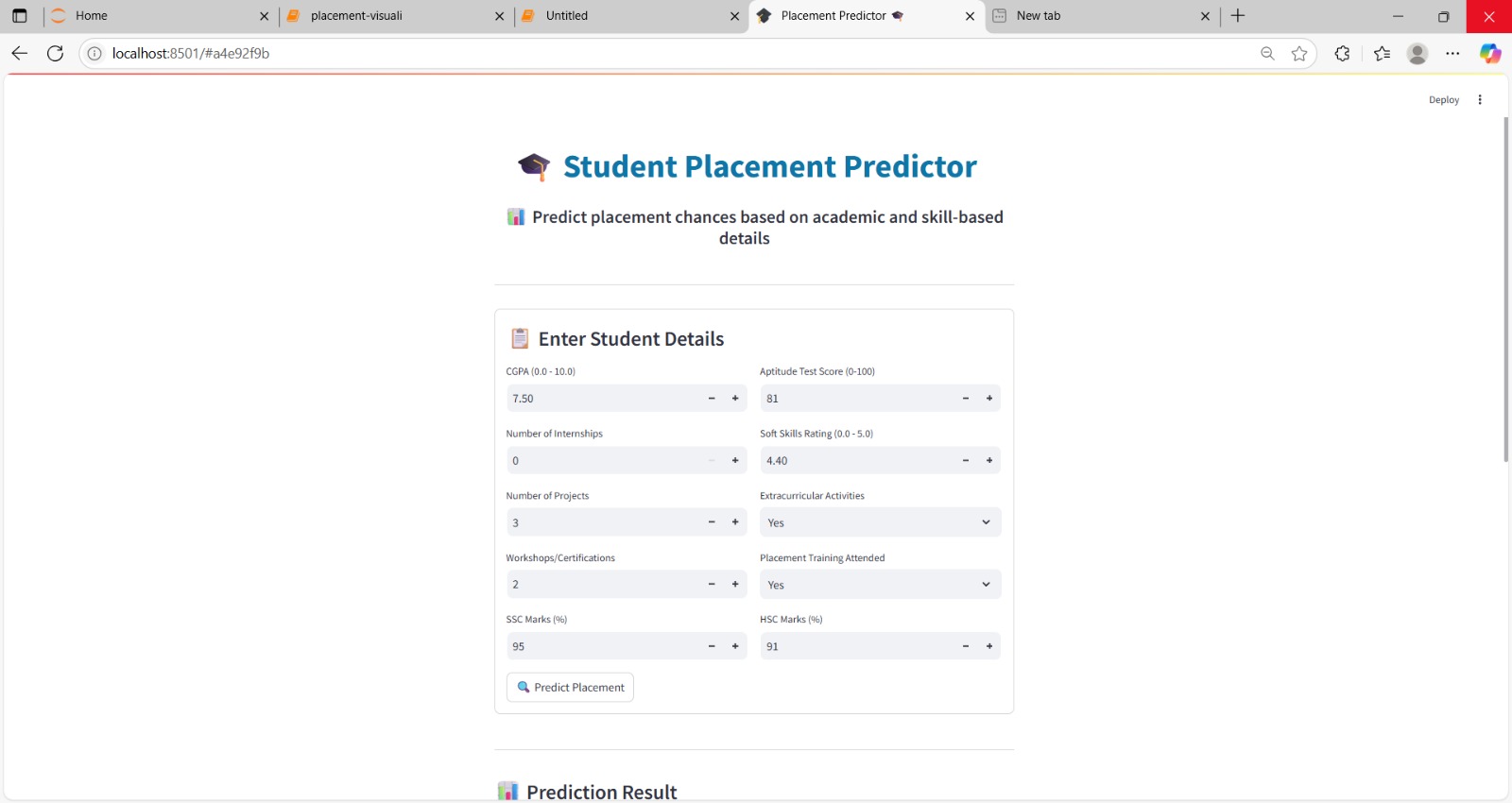
* + User input :

A screenshot of a computer

AI-generated content may be incorrect.

* + After giving the user input:





5.2 Challenges faced

During the project, several challenges were encountered. The dataset initially had missing and inconsistent values that required careful cleaning and preprocessing. Handling categorical variables like training and extracurriculars needed proper encoding for model compatibility. Ensuring a balanced distribution of placed and not placed classes was also important to avoid biased predictions. Selecting the best-performing model required multiple evaluation rounds, and integrating it into Streamlit involved aligning user input formats with the trained model’s expectations. Despite these challenges, the end-to-end pipeline was successfully implemented and deployed.

5.3 Highlights   
Based on gathering data and preliminary processing to model training, evaluation, and deployment, the entire assignment offered invaluable practical expertise in the full machine learning lifecycle. We acquired hands-on experience administering datasets from the real world, conducting data exploration, and resolving complications such as class imbalance and missing data. By categorisation models like Random Forest, XGBoost, and Logistic Regression, we had the opportunity to acquire a deeper grasp of method selection and performance modification. Employing Streamlit to implement the model also taught us how to develop engaging web-based applications that offer the  consumers with machine learning information. Everything being taken into account, the project strengthened our technical skills, intellectual, and capacity to resolve issues in the context of real life.

* 1. **Conclusion**

6.1 Summary

The Student Placement Predictor is a machine learning–powered application designed to forecast whether a student is likely to be placed based on key academic, technical, and soft skill attributes. By analyzing data such as CGPA, internships, projects, aptitude scores, and training participation, the system provides real-time predictions and personalized suggestions to boost employability.

Built using classification algorithms like Logistic Regression, Random Forest, and XGBoost, the model achieved an accuracy of over 78%. The entire solution is deployed as an interactive web app using Streamlit, offering a seamless and accessible interface for students and faculty.

A key highlight of this project is the Placement Readiness Score (0–100), which offers a measurable insight into a student’s overall employability. Along with tailored feedback, it serves as a smart, student-focused solution that connects potential with opportunity.

More than just a prediction engine, this system functions as a personalized career development tool for students and a strategic resource for institutions committed to improving placement outcomes.