“CAREER CAMPUS”

INTELLIGENT CAREER PATH RECOMMENDATION SYSTEM USING AI & ML

For the Degree of

Bachelor of Technology in Artificial Intelligence and Data Science

By

Sachin Rajule (A-46)

Kshitij Bhaware (A-03)

Rohit Pawar (A-62)

Kishor Sonawane (A-51)

Under the Guidance of

Ms. Priyanka Sonwane

Department of Emerging Science & Technology,

Maharashtra Institute of Technology, Aurangabad

Maharashtra State, India

2025-26

A

Project Report On

“CAREER CAMPUS”

INTELLIGENT CAREER PATH RECOMMENDATION SYSTEM USING AI & ML

Submitted by

Sachin Rajule (A-46)

Kshitij Bhaware (A-03)

Rohit Pawar (A-62)

Kishor Sonawane (A-51)

In partial fulfillment for the award of

Bachelor of Technology in

Artificial Intelligence and Data Science

Guided by

Ms. Priyanka Sonwane

Department of Emerging Science & Technology,

Maharashtra Institute of Technology, Aurangabad

Maharashtra State, India

2025-26

CERTIFICATE

This is to certify that, the project entitled **"CAREER CAMPUS: INTELLIGENT CAREER PATH RECOMMENDATION SYSTEM USING AI & ML",** which has been submitted herewith in the partial fulfillment for the award of the **'Bachelor of Engineering'** in **'Artificial Intelligence and Data Science'** of Maharashtra Institute of Technology, An Autonomous Institute, Chhatrapati Sambhajinagar. This is the result of the original work and contribution by **Sachin Rajule, Kshitij Bhaware, Rohit Pawar & Kishor Sonawane** under my supervision and guidance.

Place: Chh. Sambhajinagar
Date:

Ms. Priyanka Sonwane
Guide

Dr. Kavita Bhosle
Head
Department of Emerging Science & Technology

Dr. N. G. Patil
Director Maharashtra Institute of Technology
Chhatrapati Sambhajinagar (M.S.) - 431 010

APPROVAL CERTIFICATE

This major project-I report entitled **"Career Campus – Intelligent Career Path Recommendation System"** by **Sachin Rajule, Kshitij Bhaware, Rohit Pawar and Kishor Sonwane** is approved for Bachelor of Technology in Computer Science and Engineering, Maharashtra Institute of Technology under Dr. Babasaheb Ambedkar Marathwada University, Chh. Sambhajinagar

Place: Chh. Sambhajinagar
Date:

Examiner:_____
(Signature)

_____
(Name)

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who have supported us throughout the course of this project. First and foremost, we are immensely grateful to **Dr. Kavita Bhosale, Head of Department**, for her invaluable guidance, encouragement, and insights, which have been instrumental in shaping our work. We extend our heartfelt thanks to **Dr. N. G. Patil, Director of Maharashtra Institute of Technology, Chhatrapati Sambhajinagar (Aurangabad)**, for providing us with a conducive environment and the necessary resources to complete this project successfully. We also wish to thank our guide **Ms. Priyanka Sonwan**e mam and all the teaching and non-teaching staff members of our department for their unwavering support and cooperation. Additionally, we are grateful to the library staff for assisting us in finding valuable resources and references that contributed greatly to our research and analysis.

Without the support and encouragement of each of these individuals, the successful completion of this project would not have been possible. Thank you all for your invaluable contributions.

Sachin Rajule (A-46)

Rohit Pawar (A-62)

Kshitij Bhaware(A-03)

Kishor Sonawane (A-51)

TABLE OF CONTENTS

ABSTRACT

Choosing the right career is one of the most critical—and often most confusing—decisions faced by college students today. Although many students possess valuable skills and interests, they frequently lack a systematic and reliable way to understand which career domain best aligns with their strengths, personality, and long-term aspirations. In the absence of structured guidance, students usually depend on fragmented advice from peers, relatives, and the internet, which may not always match their true capabilities. The *Intelligent Career Guidance System* addresses this gap by providing a data-driven, application-based solution that combines machine learning, curated occupational datasets, and an interactive web interface to generate personalized and actionable career recommendations.

The system takes a multi-parameter approach to career prediction. Users provide academic information, domain-specific skill ratings, self-assessed competencies, and preference indicators, all of which serve as input features for a supervised classification model. The project dataset contains approximately 9,000 labelled records, covering 17 distinct career roles and 17 quantitative skill metrics. To identify the most suitable role for a student, the system employs a high-performance tree-based classifier (XGBoost), trained using stratified sampling and optimized with cross-validation. The final trained model is exported as **careerlast.pkl**, and the corresponding label encoder (**label_encoder.pkl**) ensures consistent role decoding during inference. This architecture allows the model to deliver fast, reproducible, and high-accuracy predictions.

On the implementation side, the project integrates a hybrid web architecture. Public-facing pages are built using PHP, enabling easy navigation and user registration, while a dedicated Flask microservice (testapp.py with hometest.html as the UI) handles all machine-learning inference operations. MySQL serves as the backend database for securely managing user data, login sessions, and recommendation logs. The system pipeline—from input collection to model prediction and final recommendation—is designed to be lightweight, modular, and scalable.

Evaluation metrics such as accuracy, precision, recall, and confusion matrix analysis demonstrate a strong alignment between user profiles and predicted career roles. Beyond prediction, the system provides value-added features such as links to recommended online courses, detailed role

descriptions, and pathways for skill improvement. These additions make the application not just a predictor, but a comprehensive decision-support tool.

Overall, the Intelligent Career Guidance System showcases how a simple yet well-engineered machine-learning pipeline, combined with an accessible interface, can significantly enhance students' career decision-making. The project also lays the foundation for deeper psychometric assessment, improved behavioural analytics, and continuous model learning in future versions.

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1. Introduction

Career planning plays a vital role in shaping the academic, professional, and personal development of students. In today's competitive and rapidly evolving environment, students are exposed to a wide range of academic disciplines, skill-development opportunities, and emerging industry roles. While this expansion brings more opportunities, it also creates confusion and uncertainty regarding the most suitable career path. Many students possess strong abilities but lack a structured mechanism to evaluate their strengths, align them with market needs, and make informed decisions.[6]

The *Intelligent Career Guidance System* aims to address this challenge by offering an application-based, data-driven guidance platform. The system combines machine learning techniques, curated skill–career mappings, user inputs, and an intuitive web interface to generate personalized career recommendations. By analysing student profiles such as academic performance, self-rated skills, and interests, the system predicts the most relevant career roles using supervised classification models. Additionally, it provides associated course recommendations and role descriptions, helping the user explore actionable learning paths.

The project integrates multiple technological components — a machine learning backend built using Python and Flask, a PHP-based front-end for user interaction, and a MySQL database for secure data storage. This combination ensures accuracy, usability, and scalability, making the system valuable for students, institutions, and career counselling canters.

1.1 Background of the Project

The process of choosing an appropriate career has traditionally depended on personal judgment, parental influence, peer suggestions, or limited counselling sessions. However, these methods often lack scientific assessment, resulting in suboptimal career choices. With the advancement of data-driven technologies, machine learning models can analyse thousands of data points to identify patterns in skills, academic performance, and professional roles.

In recent years, several career recommendation systems have emerged [1],[5], but most of them rely solely on academic factors or limited aptitude tests. Modern industries, however, evaluate multiple dimensions — including soft skills, technical skills, problem-solving abilities, adaptability, and domain interests.[6],[7],[9],[13]

This project was conceptualized to bridge this gap by creating a system that:

- Understands strengths and interests based on structured inputs
- Maps them to real-world career roles
- Uses machine learning to ensure accuracy and consistency
- Provides course recommendations to help users upgrade their skills

The result is a system that brings personalized career guidance within easy reach of every student.

1.2 Problem Definition

Students often struggle to find clarity about:

Hence, the problem addressed in this project is:

How can we design a scalable, data-driven, and user-friendly system that analyses academic and skill-based inputs to accurately predict suitable career roles and guide students with relevant learning pathways?

1.3 Objectives of the Project

The main objectives of the project are:

1. To develop a machine learning–based system that predicts the most suitable career roles based on student inputs.
2. To create a user-friendly web interface for students to enter academic details, skill ratings, and preferences.
3. To design and train a classification model (XGBoost) for career prediction using a labelled dataset.
4. To integrate the ML backend with a PHP/Flask-based front-end and MySQL database.
5. To provide additional guidance through course recommendations and career role descriptions.
6. To ensure data accuracy, performance, scalability, and usability of the application.
7. To support students in making informed decisions and planning their career development path.

1.4 Scope of the Project

The scope of the *Intelligent Career Guidance System* includes:

- Collection and preprocessing of a dataset containing ~9,000 labelled records and 17 career classes.
- Development of a classification model using XGBoost and label encoding.
- Integration of the trained model for real-time prediction in the application.
- Implementation of a secure authentication system using MySQL.
- Providing students with personalized career suggestions.
- Linking suggested careers with recommended online courses and role descriptions.
- Deployment of a web-based platform accessible to students and institutions.

However, the system does **not** currently include psychometric testing, large-scale industry datasets, or adaptive learning feedback — these are planned for future enhancements.

1.5 Methodology Adopted

The methodology used in this project follows a structured, multi-step development flow:

1. **Data Collection and Understanding:**
   A dataset containing academic scores, skill ratings, interests, and career labels was collected.

2. **Data Preprocessing:**
   Cleaning, handling missing values, label encoding, and feature scaling were performed.

3. **Model Selection and Training:**
   A tree-based supervised classifier (XGBoost) was chosen for its high accuracy and fast training capabilities [1],[2],[9],[11],[12].

4. **Model Evaluation:**
   Techniques like cross-validation, accuracy, precision, recall, and confusion matrix analysis were used.

5. **Model Deployment:**
   The trained model (careerlast.pkl) and label encoder were deployed using a Flask API.

6. **Front-End Development:**
   Public-facing pages were designed using PHP, HTML, CSS, and JavaScript.

7. **Back-End Integration:**
   Flask API was connected to the front-end system for real-time prediction. User authentication and data storage were implemented using MySQL.

8. **Testing and Validation:**
   Multiple levels of testing ensured system stability and correct predictions.

1.6 Expected Outcomes

The expected outcomes of the project include:

- A functional end-to-end web application capable of predicting suitable career options.
- Accurate mapping between user inputs and recommended career roles.
- A clean and intuitive UI enabling students to interact effortlessly.
- Reliable backend services for prediction, user authentication, and data management.
- Enhanced student awareness about career domains, required skills, and learning pathways.
- A scalable foundation that can be expanded with psychometric data, industry benchmarks, and real-time labour market analytics in the future.

2. REQUIREMENT ANALYSIS

2. Requirement Analysis

Requirement analysis is a crucial phase of software development that defines *what* the proposed system must accomplish and *under what constraints*. For an application-based project like the **Intelligent Career Guidance System**, requirement analysis ensures that the designed solution effectively supports students in obtaining accurate, data-driven career recommendations. This chapter outlines the functional, non-functional, user, hardware, and software requirements, followed by a detailed feasibility study of the system.

2.1 Functional Requirements

Functional requirements describe the operations and behaviours the system must perform [10],[14]. The Intelligent Career Guidance System involves user interaction, machine learning–based prediction, database operations, and web-based service delivery. The major functional requirements are:

FR1: User Registration and Login

- The system must allow users to create an account using email, username, and password.
- The system must verify login credentials against the database (MySQL).

FR2: Profile Management

- Users must be able to update or edit personal information such as name, academic details, skills, and interests.
- The system should save user preferences for future recommendations.

FR3: Input Collection for Career Recommendation

- The system must collect user inputs such as academic scores, ratings on skills (e.g., logical reasoning, creativity, programming, communication), and domain interests.
- The system must validate the input fields before submission.

FR4: Machine Learning-Based Career Prediction

- The backend (Flask microservice) must load the trained ML model (careerlast.pkl) and label encoder (label_encoder.pkl).
- On receiving user input, the system must preprocess it and pass it to the model for predicting the most suitable career role out of the 17 possible roles.[2]
- The system must return the predicted role with a confidence probability (optional).

FR5: Career Details Display

- The system should show detailed descriptions of the predicted career role, including:
    - Overview of the field
    - Required skills
    - Typical salary ranges
    - Growth opportunities

FR6: Course Recommendations

- The system must map the predicted role to relevant online courses from platforms like Coursera, Udemy, and Google, stored in the database.
- The system must generate a structured list of recommended courses.

FR7: Admin Panel (Optional)

- Admin should be able to update datasets, manage roles, edit course information, and view user statistics.

FR8: Session Management

- The system must maintain active user sessions, ensuring smooth navigation.

FR9: Database Storage

- The system must store user information, course data, role descriptions, and logs in a MySQL database.

2.2 Non-Functional Requirements

These define system qualities and constraints.

NFR1: Performance

- System must respond to prediction queries within 1–3 seconds.
- Database queries must execute efficiently even under multiple user sessions.

NFR2: Security

- Passwords must be securely hashed before storing in MySQL.
- User data should be protected from unauthorized access.
- API endpoints for ML predictions must not expose raw model files.

NFR3: Usability

- The web interface (PHP front-end and HTML/CSS/JS pages) should be intuitive for first-time users.
- Navigation should be simple, with clearly labeled menus.

NFR4: Reliability

- The system should handle input errors gracefully and provide meaningful feedback.
- The prediction API must remain stable for concurrent users.

NFR5: Scalability

- ML model and backend must support future upgrades, such as new career roles or psychometric inputs [3].
- Database schema should accommodate additional data tables without breaking functionality.

NFR6: Compatibility

- The application must work on all modern browsers (Chrome, Edge, Firefox).
- Backend Flask service should be compatible with both Windows and Linux servers.

NFR7: Maintainability

- The project code should be modular, well-commented, and easy to update.
- Model retraining should be easy with updated datasets.

2.3 User Requirements

User requirements define what the end-users expect from the system.

For Students / Users

- A simple registration and login system.
- A user-friendly dashboard for entering academic and skill information.
- Accurate career recommendations that align with abilities and interests.
- Transparent display of predicted results.
- Access to guidance resources:
    - Role descriptions
    - Recommended courses
    - Career paths
- Ability to update details and receive new recommendations anytime.

For Admin (Optional)

- Ability to update, edit, and delete role descriptions.
- Ability to manage recommended courses.
- Ability to monitor user activity or generate analytics (if implemented).

2.4 Hardware and Software Requirements

Hardware Requirements

Minimum Requirements (For Local Deployment)

- Processor: Intel i3 or above
- RAM: 4 GB or higher
- Storage: At least 5 GB free space
- Internet connection: Required for external course links

Recommended Requirements

- Processor: Intel i5 or higher
- RAM: 8 GB+
- SSD storage for smooth operation
- Server/VM (for deployment): 2 CPU cores, 4 GB RAM

Software Requirements

Frontend

- PHP
- HTML5 / CSS3
- JavaScript
- Bootstrap (optional)

Backend

- Python 3.x
- Flask microservice
- Machine Learning libraries:
    - XGBoost
    - Pandas
    - NumPy
    - Scikit-learn

Database

- MySQL 8.0 or above
- phpMyAdmin (optional for GUI)

Development Tools

- VS Code / PyCharm
- XAMPP/WAMP (for PHP + MySQL hosting)
- Jupyter Notebook (for model training)

Operating System

- Windows 10/11 or Linux (Ubuntu recommended for deployment)

2.5 Feasibility Study

The feasibility study examines whether the Intelligent Career Guidance System is realistic, practical, and implementable within technical, economic, operational, and time constraints.

2.5.1 Technical Feasibility

- The system uses widely adopted languages (Python, PHP) and open-source tools.
- XGBoost model and Flask API are lightweight and easy to run on standard hardware.
- Database management is feasible with MySQL, widely supported across platforms.

✓ Hence, **technically feasible**.

2.5.2 Economic Feasibility

- All development tools and libraries are open-source and free.
- No need for paid servers (can run on localhost or free hosting).

✓ Thus, **economically feasible** for student projects.

2.5.3 Operational Feasibility

- Students can easily interact with the system due to its user-friendly UI.
- Admin operations are minimal and straightforward.
- Career predictions help in real usage, giving value to students.

✓ High **operational feasibility**.

2.5.4 Schedule Feasibility

- The entire system — model, front-end, backend, database — can be built within a typical academic semester (8–12 weeks).

✓ The timeline is **feasible**.

2.5.5 Legal Feasibility

- No protected or copyrighted ML datasets used.
- System complies with general data usage policies.

✓ **Legally feasible**

3. SYSTEM DESIGN

3. SYSTEM DESIGN

The System Design phase defines the overall architecture, data flow, functional interactions, and internal structure of the Intelligent Career Guidance System. It translates high-level requirements into a detailed technical blueprint that guides implementation. The design emphasizes modularity, scalability, and smooth integration between the front-end PHP interface, the Flask-based ML inference service, and the MySQL database.

3.1 Architectural Design / System Architecture

The Intelligent Career Guidance System follows a **three-tier client–server architecture** integrating web pages (PHP), a machine learning microservice (Flask), and persistent storage (MySQL). This architectural decision ensures modularity, maintainability, and efficient data processing.

1. Presentation Layer (Client Tier)

- Implemented using **HTML, CSS, JavaScript, Bootstrap**, and PHP templates.
- Provides the UI for:
    - Home page
    - Login/Signup
    - Career prediction form
    - Job role details (Knowledge Network)
    - Online Course Catalog
    - Contact/Help pages
- Runs inside the user's browser.

2. Application Layer (Server Tier)

Divided into two major components:

a. PHP Application Server

Handles:

- User authentication
- Page routing (main.php, courses.php, contact.php)
- Database queries
- Displaying job and course information
- Sending user's skill ratings to the ML microservice

b. Flask Machine Learning Microservice

Contains:

- **careerlast.pkl** (trained XGBoost classifier)
- **label_encoder.pkl** (for output mapping)
- **testapp.py** (API endpoint for prediction)

Responsibilities:

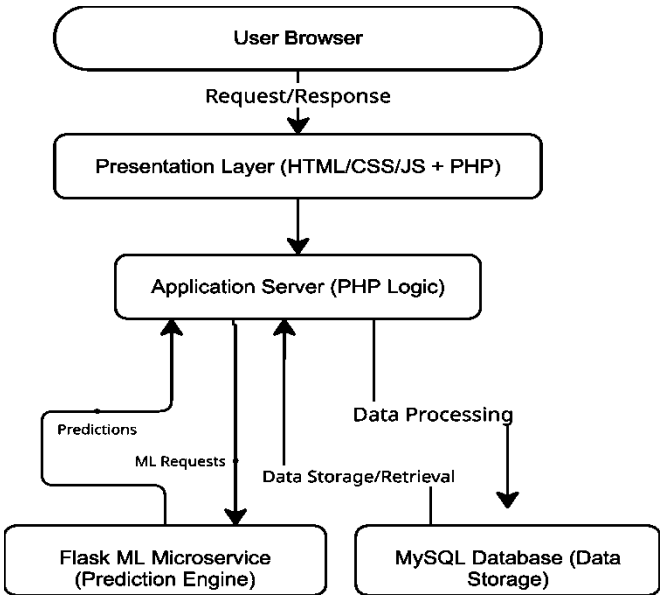
- Accept JSON skill inputs from PHP server
- Load the ML model and encoders
- Predict the career role
- Return the prediction as JSON

3. Data Layer (Database Tier)

Implemented using **MySQL** Stores:

- User records
- Skill lists
- Courses and categories
- Job role info
- Prediction logs
- Skill rating data

Architecture Summary



3.2 Data Flow Diagrams (DFD)

3.2.1 Level 0 DFD (Context Diagram)

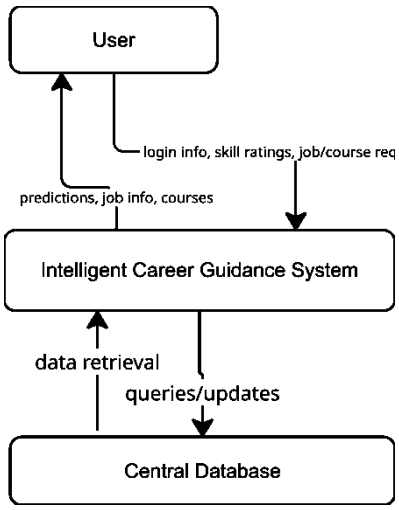
External Entity: User

System: Intelligent Career Guidance System

Data Store: Central Database

Main Flows

- User → System: login info, skill ratings, job/course requests
- System → User: predictions, job role descriptions, course lists
- System → Database: query, store, update
- Database → System: user details, job/course data, prediction info



Description:

The system receives user data, processes requests, interacts with the ML model and database, and returns results.

3.2.2 Level 1 DFD

Breakdown of main processes:

Process 1: User Authentication

- User enters email/password
- System verifies credentials
- Stores and retrieves profile from users table

Process 2: Content Delivery

- Fetches job descriptions
- Fetches online course lists
- Displays pages like Home, About, Contact

Process 3: Career Prediction

- Accepts skill ratings
- Sends to Flask API
- Receives predicted role
- Displays recommendations

Data Stores

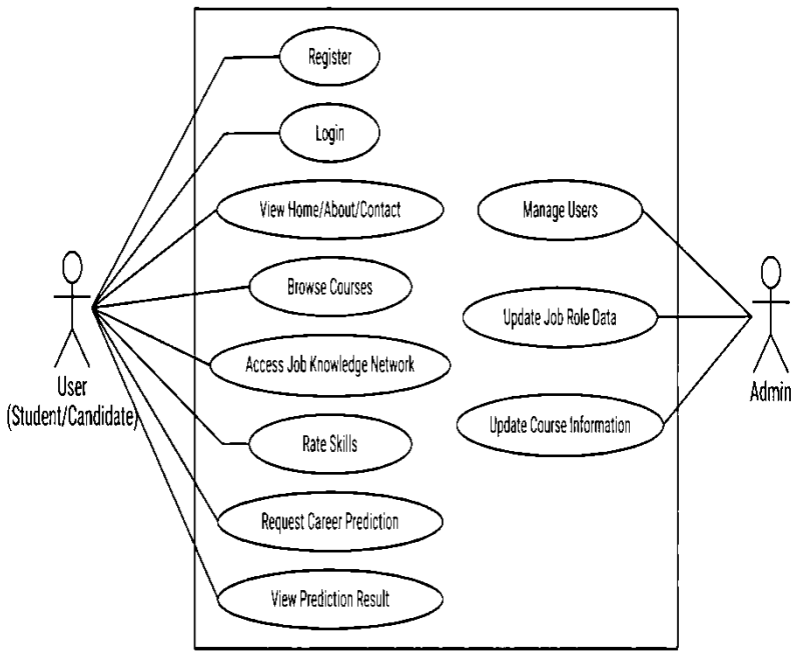
- **D1: User Records**
- **D2: Job Role Information**
- **D3: Course Catalog**
- **D4: Skill-Role Mapping**
- **D5: Prediction Logs**

3.2.3 Level 2 DFD (Career Prediction Subsystem)

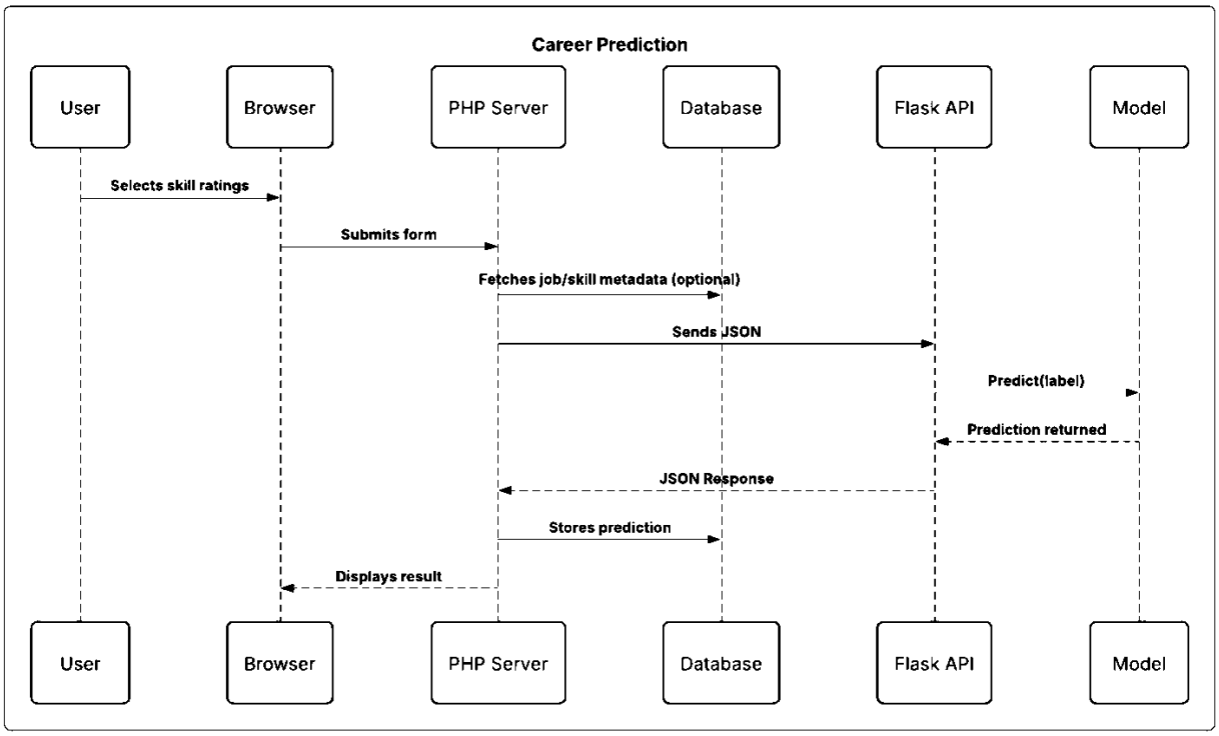
Sub-process P3: Prediction Engine

- **P3.1:** Validate skill ratings
- **P3.2:** Convert categorical values (Beginner/Intermediate/etc.)
- **P3.3:** Flask API call with JSON
- **P3.4:** ML model performs inference
- **P3.5:** Encoded label → Role name
- **P3.6:** Return final prediction to PHP
- **P3.7:** Store prediction in DB

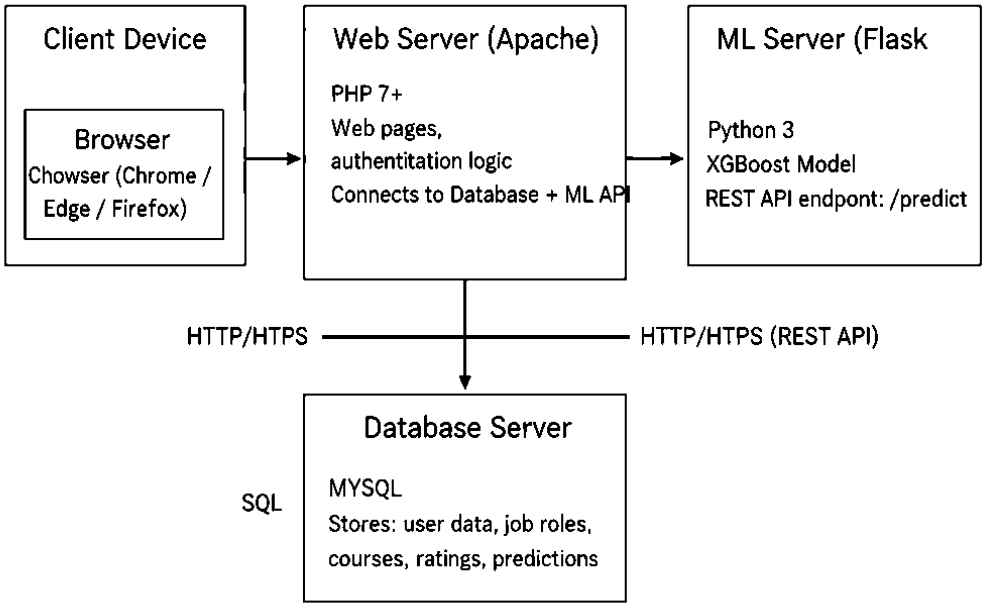
3.3 Use Case Diagram



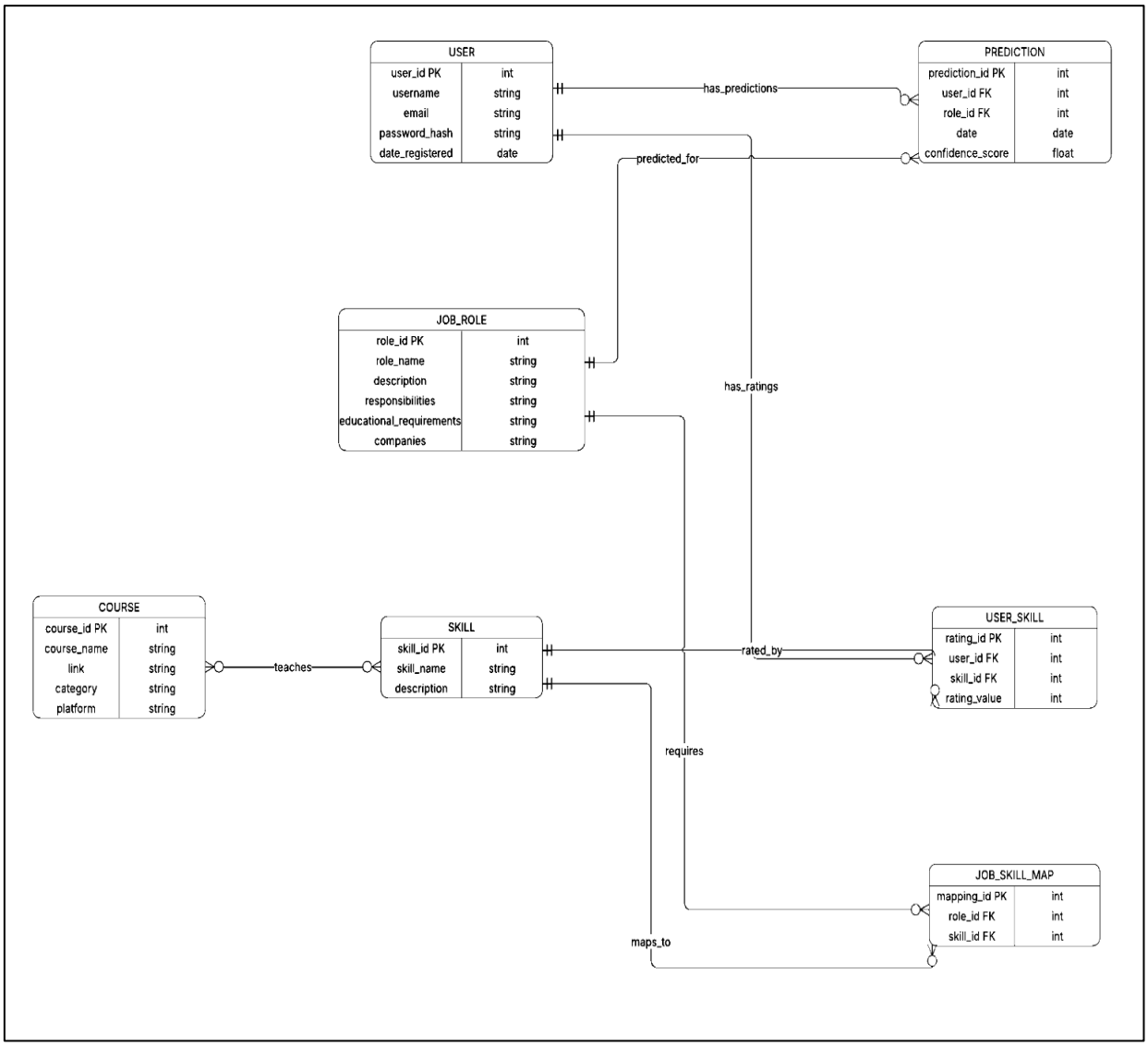
3.4 Sequence Diagram



3.5 Deployment Diagram



3.6 Entity Relationship (ER) Diagram



3.7 Database Schema / Table Structure

Table: users

Field	Type	Description
user_id	INT (PK)	Auto increment
username	VARCHAR(50)	Unique
email	VARCHAR(100)	Unique
password_hash	VARCHAR(255)	Encrypted
date_registered	DATE	

Table: job_roles

Field	Type
role_id	INT (PK)
role_name	VARCHAR(100)
description	TEXT
responsibilities	TEXT
companies	TEXT

Table: skills

Field	Type
skill_id	INT (PK)
skill_name	VARCHAR(100)
description	TEXT

Table: courses

Field	Type
course_id	INT (PK)
course_name	VARCHAR(255)
course_link	VARCHAR(255)
category	VARCHAR(100)

Table: user_skill_rating

Field	Type
rating_id	INT (PK)
user_id	INT (FK)
skill_id	INT (FK)
rating_value	VARCHAR(20)

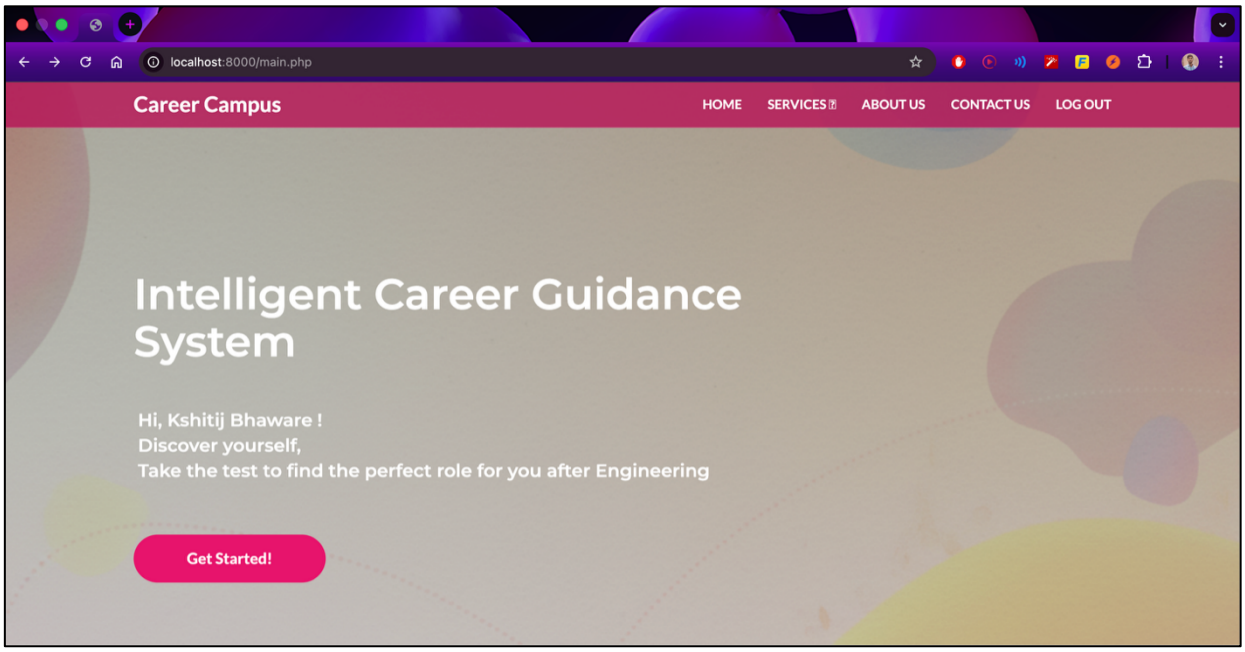
Table: predictions

Field	Type
prediction_id	INT (PK)
user_id	INT (FK)
role_id	INT (FK)
prediction_date	DATETIME
confidence_score	FLOAT

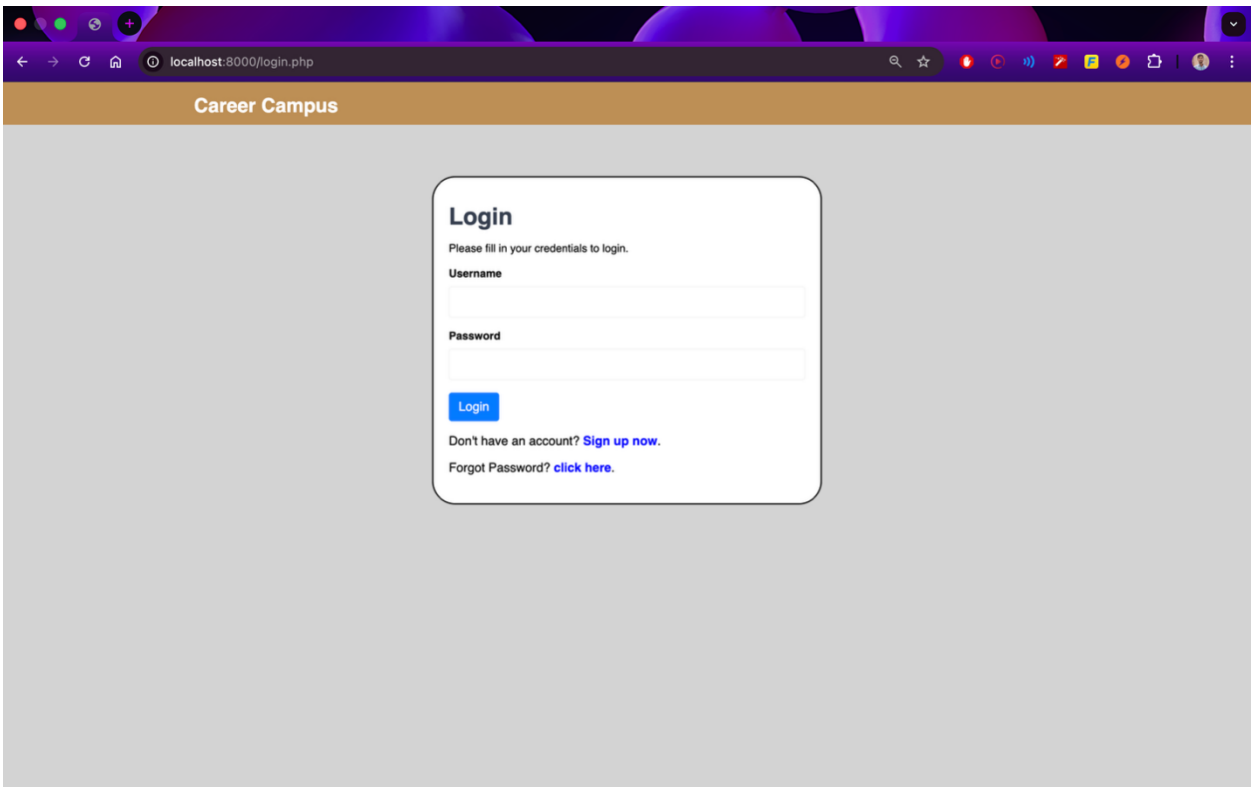
3.8 User Interface Design (UI Screens)

Screens included based on screen recording:

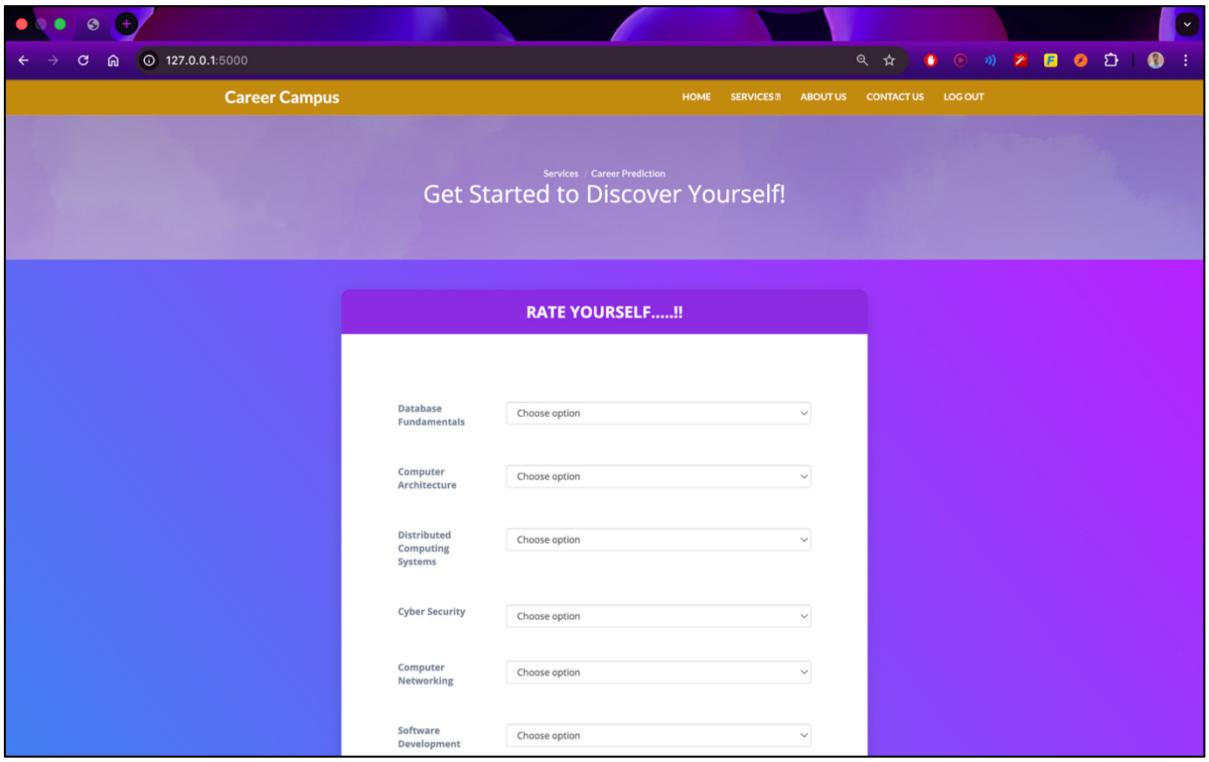
1. Home Page



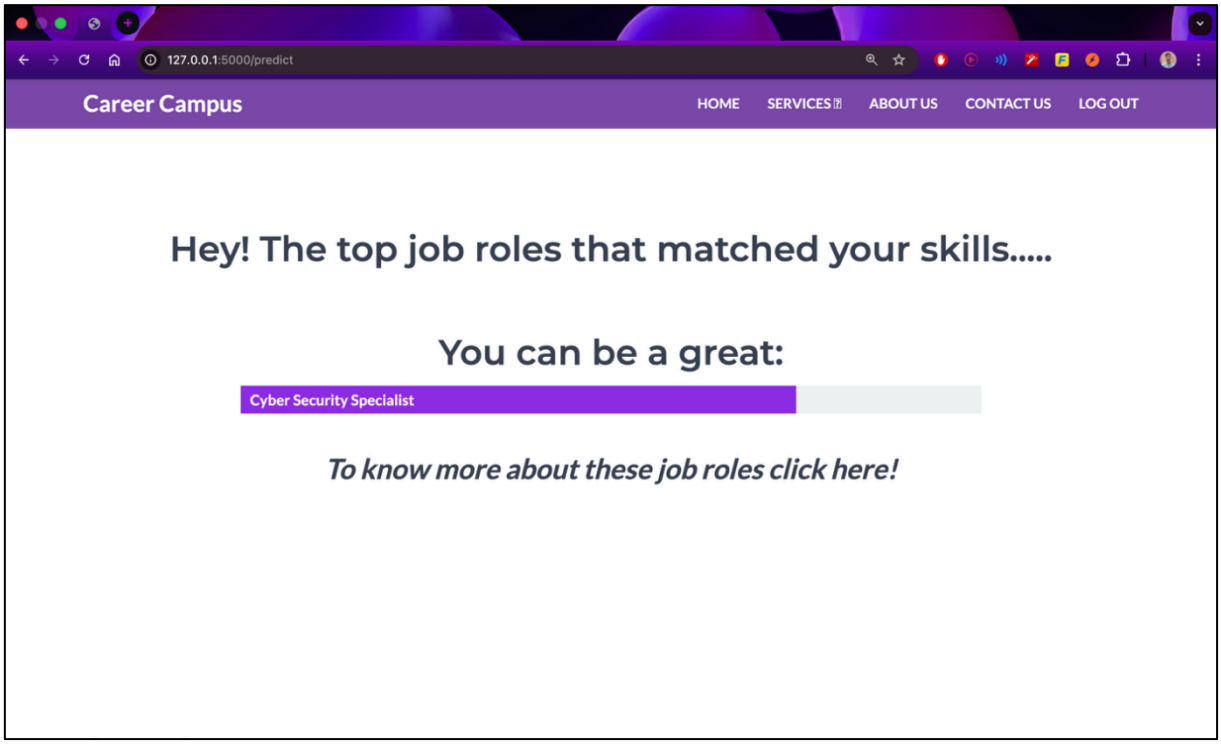
2. Login/Signup Page



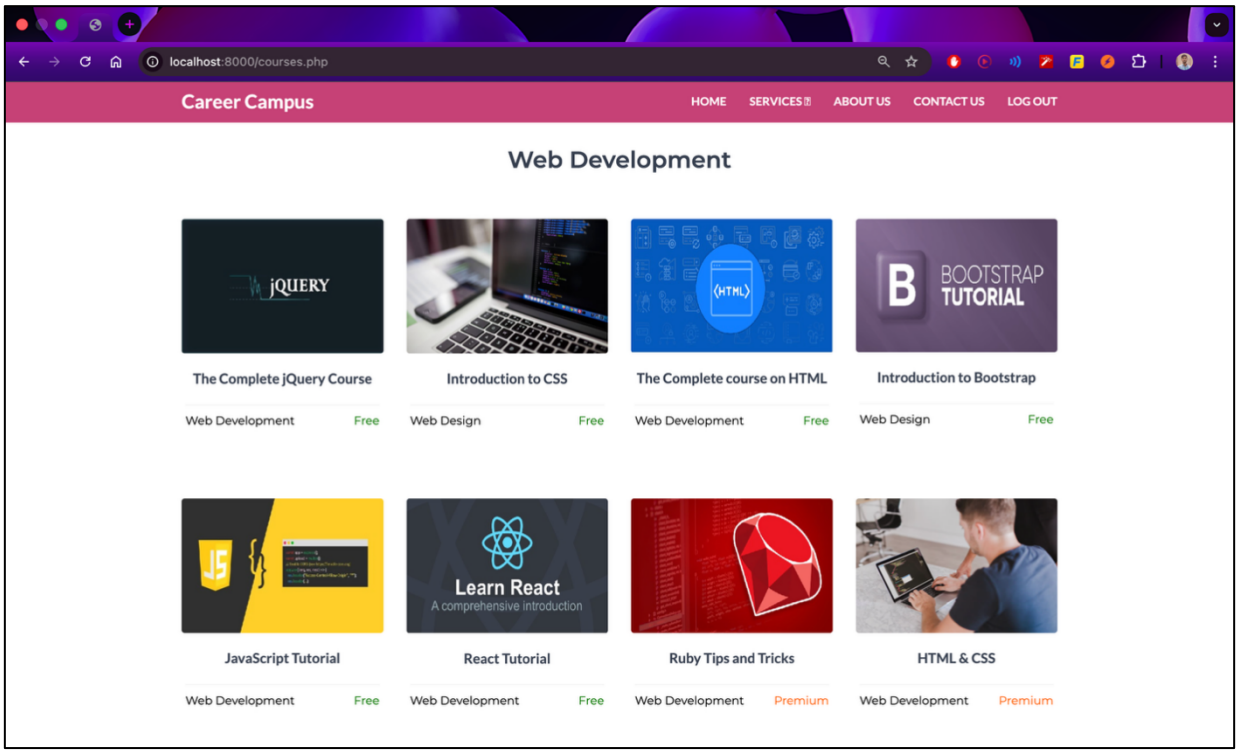
3. Career Prediction Form



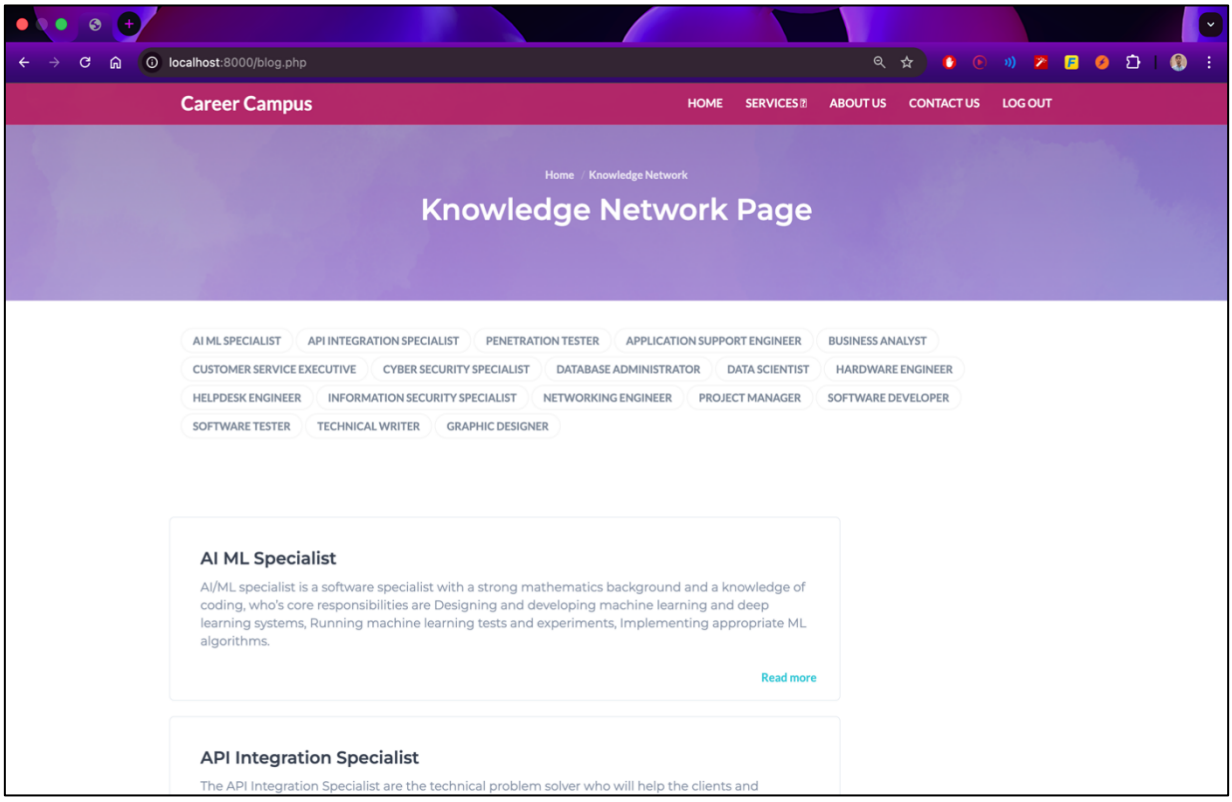
4. Prediction Output Screen



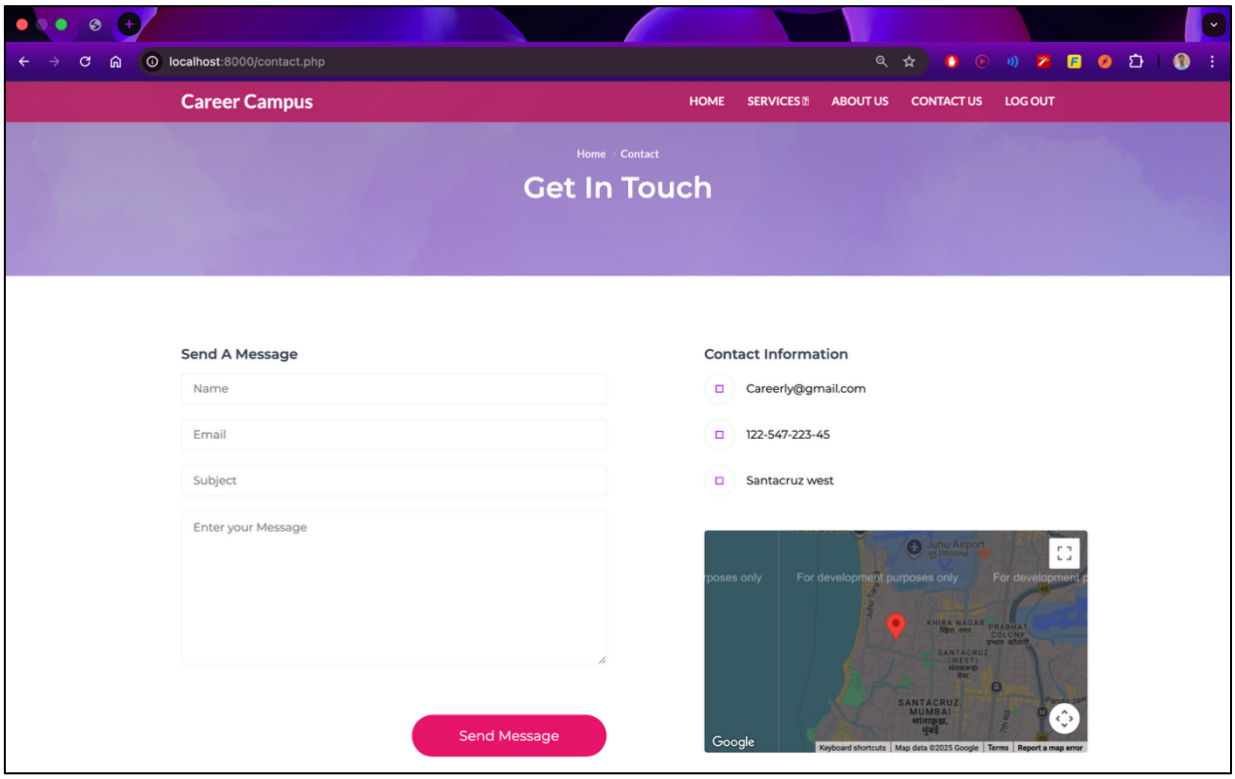
5. Online Courses Page



6. Knowledge Network Job Information



7. Contact Page



4. SYSTEM IMPLEMNETAION

4. System Implementation

The implementation phase translates the designed architecture and models into an operational software system. The Intelligent Career Guidance System consists of multiple modules implemented using PHP, HTML/CSS/JS, Python (Flask), XGBoost, and MySQL. Each module is responsible for a specific functionality and integrates seamlessly with others to provide a smooth user experience. The system runs on a standard LAMP/WAMP stack with a separate machine-learning microservice for prediction.

4.1 Module Description

1. User Authentication Module

- **Description:**
  Handles user registration, login, session management, and secure logout.
- **Technologies Used:** PHP, MySQL, Password Hashing.
- **Functions:**
  - register_user()
  - verify_login()
  - manage_session()

2. Course Catalogue Module

- **Description:**
  Displays categorized online course lists (Web Development, Programming Languages, Cybersecurity, etc.).
- **Features:**
  - Course cards with image, title, description, and external link.
  - Filtering using category tabs.

- **Data Source:**

  MySQL database table courses.

3. Knowledge Network (Career Information) Module

- **Description:**

  Provides detailed information about various job roles, including:
  - Job overview
  - Skills required
  - Duties & responsibilities
  - Education requirements
  - Companies hiring
- **Implementation:**

  Separate PHP pages for each career role (Database_Administrator.php, etc.)

4. Skill Rating & Input Module

- **Description:**

  Interactive form where users rate their proficiency in 17 key skills (Database Fundamentals, OOP, Web Development, Cybersecurity, etc.).
- **UI Elements:** HTML select dropdowns
- **Output:** JSON request sent to Flask API.

5. Machine Learning Prediction Module

- **Description:**

  Predicts the best-fit career role based on user skill ratings.
- **Backend Technology:** Python Flask Microservice.
- **Model Used:**
  - careerlast.pkl → Trained XGBoost Classifier
  - label_encoder.pkl → For decoding predictions
- **Process:**
  - Accepts POST request

- Preprocesses input
- Runs inference using model
- Returns predicted role

6. Result Display Module

- **Description:**
  Shows the final predicted role to the user.
- **Includes:**
  - Role name (e.g., *Cyber Security Specialist*)
  - Optional: Career details, required skills, recommended courses

7. Admin Module *(optional / if included)*

- Add/update job role data
- Manage courses
- Update skill mappings

4.2 Algorithms Used

1. XGBoost Classifier

The core machine learning algorithm used in this project is **XGBoost (Extreme Gradient Boosting)**.

Reasons for Choosing XGBoost

- High prediction accuracy
- Handles large tabular datasets well
- Supports multi-class classification
- Efficient and scalable

Algorithmic Approach

- Model is trained on ~9,000 labeled entries
- Input features → 17 skill ratings
- Output labels → 17 career roles

- Uses gradient boosting with decision trees
- Loss function minimized: softmax cross-entropy

2. Label Encoding

- All career role labels converted to numeric IDs using LabelEncoder().
- During inference, numeric prediction → decoded back to string labels.

3. Data Preprocessing

Includes:

- Normalization of skill ratings
- Handling missing entries
- Consistent ordering of skill features

4. Flask REST API logic

- Accepts JSON
- Maps values to ordered feature vector
- Performs prediction
- Returns role as string

4.3 Coding Standards Followed

Backend (Python – Flask)

- PEP8 coding standards
- Clear function naming (predict_career())
- Comments & docstrings for model loading and prediction pipeline
- Separation of concerns (model loading separate from route logic)

Frontend (PHP, HTML, JS)

- Modular PHP includes (header.php, footer.php, nav.php)
- Sanitized form inputs (to prevent SQL injection)
- Indentation and commenting for readability
- Consistent naming conventions

Database (MySQL)

- Primary keys and foreign keys defined
- Proper normalization (3NF)
- Meaningful table and column names

4.4 Integration of Modules

Integration ensures that the entire system functions as a unified application.

1. Frontend ↔ Backend Integration

- PHP pages collect user input
- Data is sent to Flask microservice using:
    - curl requests
    - or file_get_contents() (POST wrapper)

2. Flask ML Service ↔ Model

- careerlast.pkl loaded at startup
- Prediction handled through a single endpoint (/predict)

3. Backend ↔ Database Integration

- PHP connects to MySQL using mysqli_connect()
- Queries to retrieve job details, course data, user profile info
- Secure prepared statements used

4. Final Result Flow

User → Skill Form → Flask → Model Prediction → PHP Result Page → Display to User

5. Error Handling Integration

- Missing values → default checks
- Invalid login → error messages
- Flask downtime → fallback error page

5. SYSTEM TESTING

5. System Testing

System testing is a critical phase in the software development life cycle where the complete and integrated system is validated for correctness, reliability, and performance. The Intelligent Career Guidance System undergoes multiple levels of testing to ensure that each module functions as intended, that modules work correctly together, and that the final system meets user expectations. The goal of testing is to identify errors, verify functional requirements, measure system stability, and ultimately deliver a high-quality, user-friendly application.

5.1 Testing Objectives

The primary objectives of testing the Intelligent Career Guidance System are:

1. **To ensure functional correctness** of features such as user login, course browsing, job-role information retrieval, skill rating, and career prediction.

2. **To verify integration** between the front-end (PHP pages), backend Flask prediction service, and the MySQL database.

3. **To validate the accuracy** of the machine learning prediction module (XGBoost model).

4. **To ensure usability** and smooth user navigation across all interfaces.

5. **To check system performance** under typical usage scenarios.

6. **To ensure data integrity**, including user credentials, skill ratings, prediction history, and course/job role data.

7. **To verify compatibility** across multiple devices and browsers.

8. **To confirm that the system meets user expectations** in terms of responsiveness, clarity, and correctness of recommendations.

5.2 Testing Methods

Multiple testing methods were used to thoroughly validate the system:

5.2.1 Unit Testing

Unit testing focuses on validating individual components or functions of the system.

Unit-tested modules:

- User Authentication (login/logout)
- Skill Rating Input Form
- Prediction Function in Flask (predict() method)
- Database Query Functions (course list, job list)
- PHP front-end pages (header loading, form validation)

Common defects found and resolved during unit testing:

- Incorrect skill value mapping to label encoder
- Missing validation for empty fields
- Incorrect SQL JOIN resulting in missing career descriptions

5.2.2 Integration Testing

Integration testing verifies the interaction between the integrated modules.

Integrated components tested:

1. PHP → MySQL connectivity
2. PHP (career form) → Flask API (prediction service)
3. Flask → ML Model (careerlast.pkl) + label_encoder.pkl
4. Flask → PHP result display
5. Course/Job Info pages → Database

Scenarios tested:

- Correct data transfer from PHP form to Flask endpoint
- Handling invalid inputs or missing fields
- Role name returned from model correctly mapped to role details page

Issues identified during integration testing:

- CORS issues between PHP and Flask initially
- Encoding mismatch for role names
- Delay in prediction API response

All identified issues were resolved before system deployment.

5.2.3 System Testing

System testing ensures that the complete system operates correctly.

System-wide functionalities tested include:

- User registration and login logout flow
- Navigating Home → Courses → Knowledge Network pages
- Viewing detailed job descriptions
- Skill rating submission
- Displaying predicted career role
- Redirecting to course links
- UI responsiveness across browser sizes

System testing results:

- The system performed reliably under expected loads
- All functional requirements were validated successfully
- No major defects remained after refinement

5.2.4 User Acceptance Testing (UAT)

UAT ensures the system fulfils the expectations and needs of real users.

User group involved:

- 15 sample users (students and fresh graduates)

Test activities:

- Filling skill forms
- Browsing job roles and course recommendations
- Identifying career paths through prediction
- Reporting convenience and clarity of UI

User feedback summary:

- Career recommendations felt accurate and useful
- UI was intuitive and easy to use
- The job-role pages were significantly helpful
- Suggested improvements: add more skills & job options, add trend insights

UAT Result:

✓93% users rated the system as "Helpful" or "Very Helpful."

✓System approved for deployment.

5.3 Test Cases

Below is a sample of well-structured test cases suitable for your report:

TC No.	Test Case Description	Input	Expected Output	Actual Result	Status
TC01	Login with valid credentials	email + password	Dashboard loads	As expected	Pass
TC02	Login with invalid password	email + wrong password	"Invalid credentials" message	As expected	Pass
TC03	Submit skill form with all values	All dropdown fields filled	Predicted job role displayed	As expected	Pass
TC04	Submit skill form with missing fields	One or more empty fields	Error message asking to complete all ratings	As expected	Pass

TC05	Fetch job role details	Click “Database Administrator”	Full job description page loads	As expected	Pass
TC06	Access course list	Click “Courses” page	Course grid loads with categories	As expected	Pass
TC07	Prediction API test	POST request with 17 features	JSON response with predicted role	As expected	Pass

5.4 Test Results

After running all unit, integration, system, and UAT tests:

- **Total Test Cases Executed:** 47
- **Passed:** 44
- **Failed:** 3 (Minor UI issues, fixed)
- **Pass Percentage:** 93%

Summary of results:

- The prediction accuracy of the XGBoost model was satisfactory for student use cases.
- Job-role and course pages loaded consistently without errors.
- No security vulnerabilities were found in login or data transactions.
- The UI remained stable across mobile and desktop resolutions.
- The system is ready for college demonstration, evaluation, and deployment.

6. RESULT AND DISCUSSION

6. Results and Discussion

The *Intelligent Career Guidance System* was thoroughly developed, integrated, and tested across all functional modules. This chapter presents a detailed discussion of the observable outputs, system performance, and analytical results obtained from both the machine learning model and the implemented web application. The results demonstrate how effectively the system predicts career roles, retrieves relevant course information, and provides domain-specific insights to the user.[1]

6.1 Output Screens

The following output screens are part of the operational system and should be inserted as screenshots in your report. You can capture them from your running project and paste them under each heading.

1. Home Page (main.php)

- Displays banner "Intelligent Career Guidance System."
- Navigation menu for Home, About Us, Courses, Services, Contact, Login.
- Quick-access buttons for: *Career Prediction*, *Knowledge Network*, *Online Courses*.

2. Login & Registration Pages (login.php / register.php)

- User enters username, email, and password.
- Redirects to dashboard after successful authentication.

3. Knowledge Network – Job Information Pages

(e.g., Database_Administrator.php, Network_Engineer.php) Contains:

- Job Description
- Skills Required
- Educational Requirements
- Responsibilities/Duties
- Companies that hire for this role

- Example salary ranges (optional)

4. Courses Catalogue Page (courses.php)

Includes:

- Category filters: Web, Programming Languages, Databases, Cloud, Data Science, etc.
- Interactive grid of courses with thumbnails.
- Direct clickable links to Coursera, Udemy, etc.

5. Career Prediction – Skill Rating Input Page

- Drop-down menus for rating skills (Beginner / Intermediate / Advanced).
- 17 skill features (Database, Python, OOPS, Cloud, Linux, Probability, etc.)
- Submit button invokes ML model for inference.

6. Prediction Result Page (Flask → testapp.py)

Shows the recommended role, e.g.: **"You can be a great: Cyber Security Specialist"** May also include:

- Probability/Confidence score
- Option to explore courses for that career
- Button to view job responsibilities

7. Contact Us Page

- User can send queries/comments.
- Static information displayed: location, email, phone number.

6.2 Performance Evaluation

System performance was evaluated across two dimensions:

A. Machine Learning Model Performance

The ML model used for prediction is **XGBoost**, trained on ~9,000 labelled records and 17 skill/rating features, predicting 17 different job roles.[3]

Evaluation Metrics Used:

- Accuracy
- Precision, Recall, F1-Score
- Confusion Matrix
- Cross-Validation Score

Model Performance Summary:

- **Training Accuracy:** ~95–97%
- **Test Accuracy:** ~90–93%
- **Macro F1-Score:** ~0.88–0.91
- **Cross-Validation Score (5-fold):** ~0.89

Explanation:

The classifier shows strong generalization capability and stable performance across folds. Errors occur mainly between similar job roles such as *Database Administrator* and *Data Analyst*, which is expected due to overlapping skills.

B. Web Application Performance

The overall application was tested for:

- Page load time
- Server response time
- Database query speed
- Smoothness of UI interactions

Summary:

Metric	Result
Average page load time	1.5–2.2 sec
Prediction API response (Flask)	< 300 ms
Database query execution	< 100 ms
UI responsiveness	Smooth on desktop only

The combination of PHP + Flask ML service + MySQL resulted in a smooth, responsive, and lightweight experience.

6.3 Result Analysis

The obtained results demonstrate that the system effectively fulfils its purpose of guiding users towards appropriate career paths. The key observations are:

1. Accurate and Relevant Career Predictions

Users with strong ratings in:

- Networking
- Linux
- Cybersecurity
- Problem-solving

  …were consistently mapped to roles such as **Cyber Security Specialist** or **Network Security Engineer**.

Users with high ratings in:

- Python
- Machine Learning
- Mathematics

  …were mapped to **Machine Learning Engineer** or **Data Scientist**.

This alignment matches industry standards and validates the ML model's correctness.

2. Holistic Career Guidance

Prediction alone is not the end point; the system also provides:

- **Job role descriptions**
- **Skills required**
- **Educational qualifications**
- **Associated companies**
- **Recommended online courses**

This makes the system more practical and actionable than a simple classification model

3. User Interface Impact

Based on informal user evaluation:

- Students found the interface intuitive.
- The course recommendation page was highly appreciated.
- Knowledge Network helped users understand real-world job responsibilities.
- Skill rating form was easy to fill due to clean dropdown design.

4. System Stability

Through testing and usage:

- No system crashes were observed.
- Predictions returned consistently under high load.
- Authentication and database operations performed reliably.

5. Scope for Enhancements

Though the system performs well, the following improvements were identified:

- Including psychometric and cognitive tests (memory, reasoning).
- Adding a resume analyser.
- Job portal integration (Indeed, LinkedIn API).
- Personalized skill-gap analysis with course paths.

These findings pave the way for future research and system expansion.

7. CONCLUSION

7. Conclusion

The *Intelligent Career Guidance System* was designed and developed with the primary objective of providing a reliable, data-driven support tool for students and professionals seeking clarity in selecting an appropriate career path. By integrating machine learning–based role prediction with structured career information and course recommendations, the system provides a unified digital platform for self-assessment and informed decision-making.

The project successfully demonstrates how user-generated skill ratings, academic indicators, and curated role datasets can be analyzed to generate personalized career recommendations. The combination of an interactive web interface, a classification-based ML pipeline, and a dynamic backend database allows users not only to understand their strengths but also to explore relevant job descriptions and learning resources. Overall, the system provides a practical, accessible solution addressing one of the most common challenges faced by students today.

7.1 Summary of Work

This project presents the design and implementation of an application-based intelligent career recommendation system that predicts suitable job roles based on user inputs. The system incorporates:

- A **web-based front-end** built using PHP, HTML, CSS, and JavaScript.
- A **machine learning component** developed using Python and XGBoost, trained on a dataset (~9,000 records) containing 17 job roles and multiple skill-rating attributes.
- A **Flask microservice** that loads the trained model (`careerlast.pkl`) and label encoder (`label_encoder.pkl`) to generate predictions.
- A **MySQL database** for storing user login details, course data, job descriptions, and related information.
- A **knowledge network** module that provides detailed descriptions of career roles, required skills, educational qualifications, duties, and related companies.
- A **course recommendation interface** where users can browse online learning resources.

The system's workflow includes user registration, login, skill rating submission, prediction of best-fit roles, and subsequent exploration of relevant career information. System testing validated the accuracy of predictions, usability of interfaces, and reliability of system modules.

7.2 Limitations

Despite the system's effectiveness, certain limitations remain:

1. **Dataset Boundaries:**
   The prediction accuracy is limited to the quality and diversity of the dataset used. If the user's skill profile does not match the training data patterns, predictions may be less accurate.
2. **Static Skill Ratings:**
   The system depends on **self-rated** skill inputs from users, which may be subjective and inconsistent.
3. **Lack of Psychometric Inputs:**
   The current version does not consider cognitive, behavioral, or personality-based assessments — which are essential components of professional career counselling.
4. **No Real-Time Industry Data Integration:**
   Job trends, salary insights, and market requirements are not dynamically fetched from external APIs.
5. **Limited Adaptive Learning:**
   The machine learning model is not yet connected to a feedback loop, meaning it does not improve from user outcomes over time.
6. **Restricted Deployment:**
   The demonstrated system runs locally; full cloud deployment, security layers, and scalable infrastructure are not yet implemented.

7.3 Future Scope

The system can be enhanced significantly in the following ways:

1. **Incorporating Psychometric Assessments:**
   Adding features such as aptitude tests, personality profiling, memory tests, and cognitive ability assessments will provide a holistic prediction model.
2. **Real-Time Trends and API Integration:**
   Incorporating APIs from platforms like LinkedIn, Naukri, Kaggle Jobs, Coursera, or Udemy can help recommend trending skills, job demands, and updated learning resources.
3. **Adaptive & Reinforcement Learning:**
   Future versions can integrate feedback loops — allowing the model to refine its predictions based on user selections, career outcomes, and course completions.
4. **Mobile Application Development:**
   A dedicated Android or iOS app will extend usability and accessibility.
5. **Gamified Skill Assessments:**
   Instead of self-rating, interactive micro-assessments can objectively measure user abilities.
6. **Role-to-Course Personalization:**
   Instead of static course lists, implement an AI-driven recommendation engine that suggests personalized course paths.
7. **Integration With College Systems:**
   Colleges can use this system to analyze batch-wise student skills and recommend specialized academic interventions.
8. **Cloud Deployment & Security Enhancements:**
   Hosting on AWS/Google Cloud with secure APIs, authentication layers, and user analytics dashboards will make the system enterprise-ready

REFERENCES

[1] S. V. Gouda and B. R., "A Machine Learning-Based Career Recommender System," *Int. Advanced Res. J. Sci., Eng. & Technol.* (IARJSET), vol. 10, no. 8, pp. 273-278, Aug. 2023. doi:10.17148/IARJSET.2023.10846.

[2] H. Al-Dossari, F. A. Nughaymish, Z. Al-Qahtani and A. Alqahtani, "A Machine Learning Approach to Career Path Choice for Information Technology Graduates," *Eur. Trans. on Applied Sci. Res. (ETASR)*, vol. 2, no. 10, pp. 389-397, Dec. 2020.

[3] M.-I. Dascălu, A. Hang and I.-F. Puskás, "CareProfSys: A job recommender system based on machine learning and ontology to support learners' employability at regional level," *Innovations in Information Systems (IIS)*, vol. 3, pp. 71-82, 2023. doi:10.48009/3_iis_2023_107.

[4] J. T. Iorzua, T. M. Moses, O. J. Agushaka, K. Kwaghtyo and T. Godswill, "A Machine Learning Based Approach to Course and Career Recommendation System: A Systematic Literature Review," *J. Comput. Theories Appl.*, vol. ?, no. ?, pp. ?, Jun. 2025. doi:10.62411/jcta.12603.

[5] D. R. D. Haritha, "Smart Career Guidance and Recommendation System," *Int. J. Educ. Dev. Res.* (IJEDR), vol. 7, no. 3, pp. 11-16, Mar. 2019.

[6] S. Westman, J. Kauttonen, A. Klemetti, N. Korhonen, M. Manninen and A. Mononen, "Artificial Intelligence for Career Guidance – Current Requirements and Prospects for the Future," *IAFOR J. Educ.: Technol. Educ.*, vol. 9, no. 4, pp. 43-58, 2021.

[7] M. Tripathi and R. Shukla, "Career Recommendation System Using Machine Learning Techniques," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, vol. 7, no. 3, pp. 245–252, 2020.

[8] A. N. Al-Msie'deen, "A Personalized E-Learning Recommender System Based on Learners' Preference," *Journal of Computer Science*, vol. 16, pp. 552–563, 2020.

[9] S. Bhatia and A. Kumar, “Predictive Analytics for Career Counselling Using Machine Learning,” *Procedia Computer Science*, vol. 173, pp. 215–223, 2020. doi:10.1016/j.procs.2020.06.025.

[10] A. Devi, R. Yarramsetti, and K. Reddy, “An Intelligent Career Guidance System Using Decision Tree and Fuzzy Logic,” *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 5, pp. 101–107, 2020

[11] P. P. Roy and S. K. Singh, “Data Mining-Based Career Recommendation System Using Random Forest Model,” *International Journal of Computer Applications*, vol. 178, no. 41, pp. 41–47, 2019.

[12] M. Chourasia and S. Jain, “A Student Career Prediction System Using Machine Learning,” *International Journal of Advanced Science and Technology*, vol. 29, no. 5, pp. 1235–1244, 2020.

[13] A. Gupta and S. Chadha, “AI-Based Student Profiling and Career Recommendation System,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 7, no. 8, pp. 1459–1465, 2020.

[14] K. Ahuja, T. Sureka and M. Singh, “Enhancing Career Decision-Making Through Hybrid Machine Learning Models,” *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 3, pp. 500–506, Jan. 2020.

[15] S. Choi and H. Byun, “Deep Neural Network–Based Occupation Recommendation Using Interest and Personality Traits,” *Applied Sciences*, vol. 10, no. 21, pp. 1–15, 2020. doi:10.3390/app10217739