Chandigarh Engineering College Jhanjeri
Mohali-140307
Department of Computer Science & Engineering

Mid Term Report
on

Intelligent Career Guidance System with AI

Project-I

BACHELOR OF TECHNOLOGY
(Computer Science and Engineering)

SUBMITTED BY:
Nilesh Chhabra : 2330590
Vanshika Nain : 2230889
Renuka Rani : 2330593
Yachana Thakur : 2230898

Under the Guidance of
Dr. Ashima Shahi

Department of Computer Science & Engineering
Chandigarh Engineering College
Jhanjeri, Mohali - 140307

INDEX

Chapter No.	Description	Page No.
Chapter 1	**Introduction** This chapter must describe introduction about your project. Objective(s) of the project and Tools Learnt	**1.**
Chapter 2	**System Requirements** Software and Hardware requirements	**2.**
Chapter 3	**Software Requirement Analysis** Define the problem Define the modules and their functionalities	**3.**
Chapter 4	**Software Design** • The design part must include the following items • DFDs in case of Database projects • Database Design For database projects, • The report must include the following items. E-R Diagram.	**5.**
Chapter 5	**Implementation** Consist of coding or code outline for various files For Database projects, the report consisting of Tables – explaining all fields and their datatypes, Stored procedures (PL/SQL)	**6.**
6	**References**	**9.**

Chapter 1: Introduction

1.1 Overview

The "Intelligent Career Guidance System with AI" is a smart web-based platform that provides career recommendations to users by analyzing their academic background, interests, and aptitude. This system is designed to aid students and job seekers in making informed decisions that align with their skills and preferences. By incorporating artificial intelligence and machine learning techniques, it delivers highly personalized and relevant suggestions.

1.2 Motivation

Students often struggle to choose a career path due to limited access to personalized counseling. This is especially true in rural areas or schools without proper guidance facilities. A digital platform that offers 24/7 access to customized guidance can serve as a crucial tool in empowering youth and enabling better career planning.[1]

1.3 Objectives

- Create an intelligent system that analyzes user data using AI models.
- Offer a test interface that assesses aptitude and interest.
- Store and manage user responses in a secure database.
- Generate meaningful, visual career recommendations.
- Enable administrators to manage and update content easily.[2]

1.4 Scope

The system is scalable and can support thousands of users. In the future, it can be integrated with resume builders, college application systems, and real-time job listings. The recommendation engine can evolve by learning from user feedback and additional training data.[3]

1.5 Advantages

- Saves time compared to traditional counseling
- Provides objective, data-driven suggestions
- Easy access from any location
- Can be enhanced with NLP and chatbot interface[4]

Chapter 2: System Requirements

2.1 Software Requirements

- Operating System: Windows 10 / Ubuntu 20.04+
- Languages Used: HTML5, CSS3, JavaScript, PHP 7+, Python 3.8+
- Frameworks: Flask (Python), Bootstrap (CSS), jQuery (JS)
- Database: MySQL / MariaDB
- Server: Apache (via XAMPP/LAMP)
- IDEs: Visual Studio Code, Sublime Text [5]

2.2 Hardware Requirements

- Minimum Processor: Intel i3, Recommended: Intel i5 or above
- RAM: 4 GB (minimum), 8 GB (optimal for smooth AI processing)
- Disk Space: 1 GB for dataset, dependencies, and logs
- Input/Output Devices: Keyboard, Mouse, Monitor
- Network Interface: Required for hosting and update [6]

Chapter 3: Software Requirement Analysis

3.1 Problem Definition

Due to a lack of efficient and affordable career counseling systems, many students end up in unfit careers. This results in decreased job satisfaction and productivity. Traditional methods of career guidance are not scalable and often ignore individual uniqueness. Our system provides automated, personalized suggestions using data-driven decision-making.

3.2 Functional Requirements

- Registration and secure login/logout for users
- Conducting standardized aptitude and interest tests
- Storing and retrieving test results for each user
- AI model inference based on inputs
- Admin panel to manage question banks and career mappings
- Display career graphs and downloadable reports

3.3 Non-Functional Requirements

- High availability and fast response times
- Secure storage of user credentials and scores
- Scalable backend for concurrent users
- Responsive design for cross-platform compatibility
- Easy maintainability with modular code structure

3.4 Use Case

Use Case	**Description**
Take Test	Allows new users to create accounts
View Result	Users answer questions on aptitude and interest
Admin Login	AI predicts best-suited careers and shows results
Data Analysis	System logs prediction accuracy for future enhancement

Chapter 4: Software Design

4.1 Data Flow Diagram (DFD)

Level 0 DFD:

- External Entity: User → System → Output: Career Suggestions

Level 1 DFD:

- User submits data → PHP Handler → Python Model → MySQL → Output Module

4.2 Architecture Diagram

- **Client Tier:** Handles UI rendering and input collection
- **Server Tier:** PHP handles requests, Python processes AI logic
- **Database Tier:** MySQL for storing test data, profiles, results

4.3 ER Diagram Description

Entities:

- **User:** name, email, password
- **Test:** test_id, test_type, timestamp
- **Response:** user_id, test_id, question_id, answer
- **Career:** career_id, title, description, score_range

Relationships:

- One user can take many tests
- Each test has multiple responses
- One response links to career recommendation(s)

Chapter 5: Implementation

5.1 Frontend Design

- **Login Page**: Includes email and password validation
- **Test Interface**: Progress bar, timer, and radio buttons for MCQs
- **Dashboard**: Displays charts, personalized tips, and downloadable reports

5.2 Backend Implementation

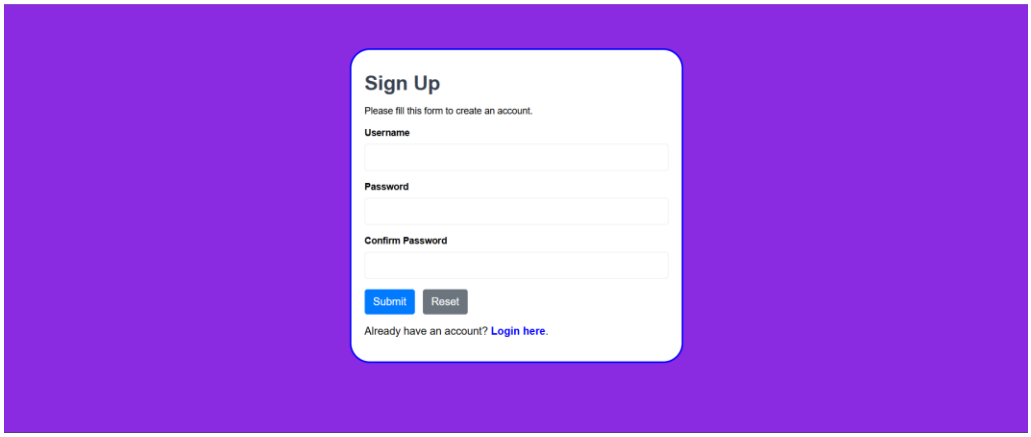
- login.php and register.php for user authentication
- submit_test.php to collect and save responses
- predict.py to generate AI predictions using scikit-learn

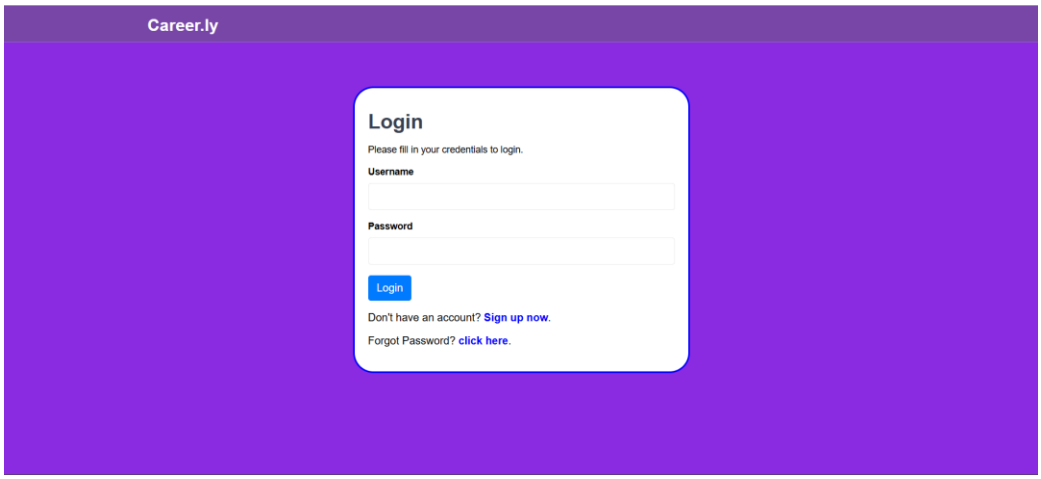
5.3 Machine Learning Pipeline

```
from sklearn.naive_bayes import GaussianNB
import pandas as pd

data = pd.read_csv('career_dataset.csv')
X = data[['aptitude', 'interest', 'academic_score']]
y = data['career']

model = GaussianNB()
model.fit(X, y)

user_input = [[8, 6, 85]]
career = model.predict(user_input)
print("Recommended Career:", career)
```

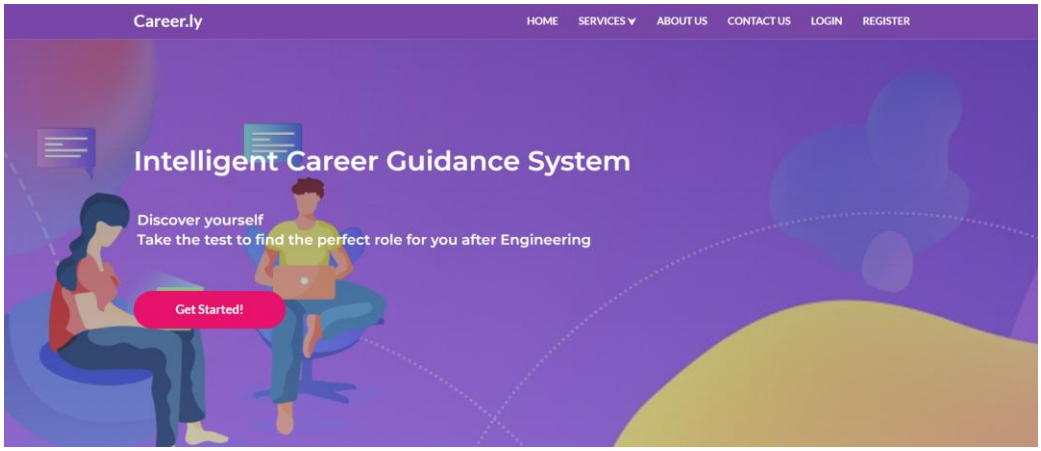
5.4 Sample Screenshots

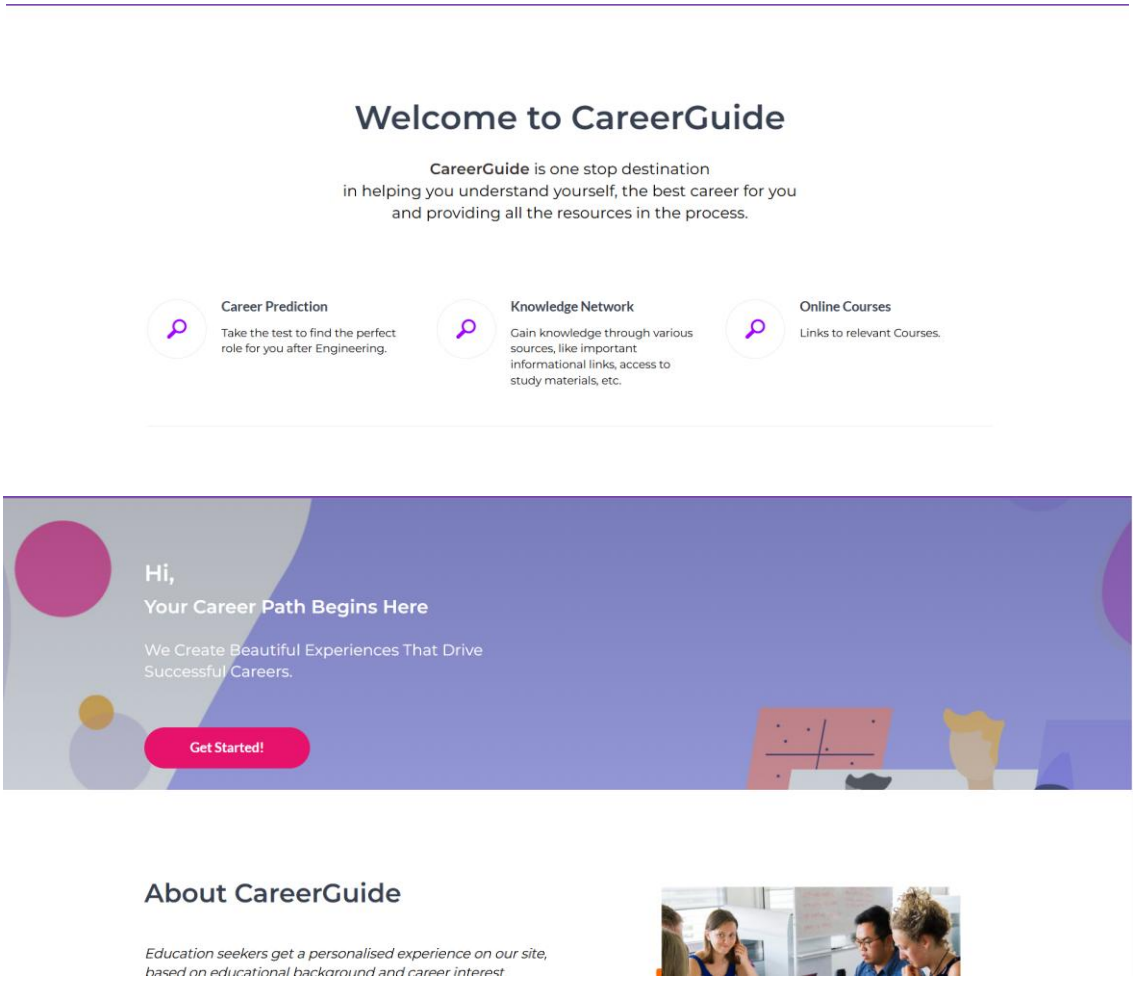
Login Page



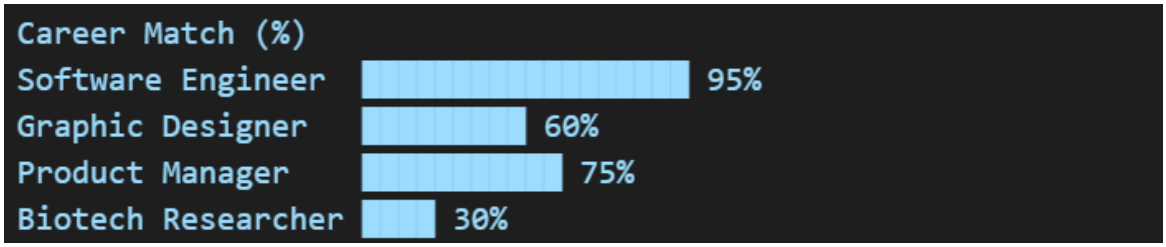


Dashboard





Career Report Graph



References

[1] Aggarwal, C.C. (2018). Data Mining: The Textbook. Springer.
[2] McKinsey & Company (2021). The Future of Work in India.
[3] scikit-learn: Machine Learning in Python – https://scikit-learn.org
[4] W3Schools – HTML/CSS/JS Reference – https://www.w3schools.com
[5] Flask Documentation – https://flask.palletsprojects.com
[6] GitHub – AI Career Recommender sample repos
[7] CareerExplorer.com – Role Descriptions & Industry Stats