ASCEND – ***A****lumni–****S****tudent* ***C****areer* ***EN****gagement* ***D****ashboard*

Problem Definition

In many higher-education institutions, junior students lack structured and reliable access to seniors and alumni who have practical industry experience. Existing communication methods such as social media groups, informal messaging, and notice boards suffer from several limitations:

- Lack of verification and accountability of career advice
- Repetitive and low-quality questions
- Mentor overload leading to disengagement
- Generic guidance that does not match a student's background
- Absence of measurable outcomes to evaluate advice quality

These issues result in inefficient knowledge transfer and poor career decision-making among students.

Proposed System

The proposed project is a **web-based mentorship and career guidance platform** developed using **Python for backend processing** and **HTML, CSS, JavaScript, and Bootstrap for frontend design**.

The system focuses on **structured interaction, fairness, and accountability**, rather than real-time chat-based communication. Mentorship is delivered asynchronously, ensuring scalability and sustainability.

Objectives

1. To provide structured and meaningful career guidance to junior students
2. To reduce mentor workload through controlled and fair question handling
3. To evaluate mentorship quality using outcome-based feedback
4. To apply **Data Structures and Algorithms (DSA)** within real system workflows
5. To design a scalable and modular web application using Python

---

System Features

1. User Roles

- **Junior Student:** Can submit structured career or technical queries
- **Senior/Alumni Mentor:** Can respond to assigned queries
- **Administrator:** Manages verification, moderation, and system integrity

---

2. Structured Question Submission

Juniors must submit queries in a predefined format including:

- Current skill set
- Target career role
- Previous attempts or preparation
- Specific question description

This ensures high-quality inputs and reduces unnecessary mentor effort.

---

3. Asynchronous Mentorship Model

- Real-time chat is avoided
- Mentors respond at their convenience
- Improves response quality and system sustainability

---

4. Outcome-Based Trust System

- Juniors provide delayed feedback on guidance received
- Trust score is calculated based on measurable outcomes
- Prevents misleading or low-quality advice

---

5. Similarity-Based Mentor Recommendation

Mentors are suggested based on:

- Academic background
- Skill level

- Career trajectory

This provides realistic and relevant guidance.

Integration of Data Structures and Algorithms

To satisfy academic and practical requirements, the project integrates DSA into core system logic:

Data Structure / Algorithm	Application in System	Purpose
Queue (FIFO)	Mentor Question Scheduling	Ensures fairness and prevents overload
Priority Queue	Urgent Career Queries	Handles deadline-based requests
Stack (LIFO)	Admin Moderation Rollback	Enables undo functionality
Hash Map	Mentor Matching	Fast lookup and efficient mapping
Sorting Algorithms	Reputation Ranking	Displays top mentors

These structures are used as part of the system's functional workflow, not as standalone demonstrations.

Technology Stack

Backend

- Python
- Flask Framework
- SQLAlchemy ORM

Frontend

- HTML
- CSS
- JavaScript
- Bootstrap

Database

- MySQL / PostgreSQL

Architecture

- MVC / Layered Architecture
- Role-based authentication

System Feasibility

- The project is modular and can be completed within a single semester
- Core features are prioritized for timely completion
- Advanced features can be implemented as extensions
- Requires no paid APIs or external services

Expected Outcomes

- A functional, secure, and scalable web application
- Demonstration of Python backend development skills
- Practical application of Data Structures and Algorithms
- Improved mentorship efficiency within an academic environment