Project Title	**Placement Eligibility Streamlit Application**
Skills take away From This Project	• **Building data-driven applications using Streamlit.** • **Generating synthetic data using the Faker library.** • **Implementing Object-Oriented Programming (OOP) principles in Python.** • **Writing and optimizing SQL queries for insights and analytics.** • **Working with relational databases to store and retrieve data.** • **Creating interactive dashboards for real-time decision-making.**
Domain	**Ed Tech**

Problem Statement

Design and implement a **Streamlit application** where users can input eligibility criteria for placement. Based on these criteria, the application should query a dataset of student information to display eligible candidates' details.

Business Use Cases

1. **Placement Management**: Filter and shortlist students based on customizable criteria.
2. **Student Performance Tracking**: Assess student readiness for placement based on key metrics.
3. **Interactive Analytics**: Provide placement teams with insights through an interactive application.

Approach

1. **Step 1**: **Dataset Creation**
    - Generate four related tables using the **Faker library**:
        - **Students Table**: Contains student information such as name, age, contact, and enrollment details.
        - **Programming Table**: Tracks student performance in programming (e.g., Python) with metrics like problems solved, assessments completed, and projects submitted.
        - **Soft Skills Table**: Tracks communication, teamwork, and presentation skill scores.
        - **Placements Table**: Contains data on placement readiness, mock interview performance, and internship experiences.
    - Establish relationships among these tables.
2. **Step 2**: **Data Storage**
    - Store the generated data in a relational database (e.g., SQLite, MySQL).
    - Implement Python OOP for database interactions.
3. **Step 3**: **Streamlit Application**
    - Create a **user-friendly interface** where users can:
        - Input eligibility criteria (e.g., problems solved > 50, soft skills score > 75).
        - View eligible students details dynamically.
4. **Step 4**: **SQL Queries and Insights**
    - Write 10 SQL queries to extract meaningful insights, such as:
        - Average programming performance per batch.
        - Top 5 students ready for placement.
        - Distribution of soft skills scores.
    - Incorporate the insights into the Streamlit app.

Results

- A fully functional Streamlit application showcasing **dynamic filtering** of eligible students.
- **10 SQL queries** providing actionable insights.
- Clean, modular code using OOP principles.

Project Evaluation Metrics

- Functionality: Application filters and displays data based on criteria.
- SQL Queries: Quality and relevance of insights extracted.
- OOP Design: Proper implementation of classes and methods.
- UI/UX: Simplicity and interactivity of the Streamlit application.
- Documentation: Completeness and clarity of the project documentation.

Technical Tags

Streamlit, **Python**, **Faker**, **SQLite**, **SQL**, **Data Science**, **OOP**, **Dashboard Development**.

Data Set

Here are the schemas for the required tables in the requested format:

Students Table

This table stores basic information about students enrolled in the course.

- **student_id (Primary Key)**: Unique identifier for each student.
- **name**: Full name of the student.
- **age**: Age of the student.
- **gender**: Gender of the student (e.g., Male, Female, Other).
- **email**: Email address of the student.
- **phone**: Contact number of the student.
- **enrollment_year**: Year when the student enrolled in the course.
- **course_batch**: Name of the batch or cohort the student belongs to.
- **city**: City of residence for the student.
- **graduation_year**: Expected or actual graduation year for the student.

Programming Table

This table stores details of students' programming performance in the course.

- **programming_id (Primary Key)**: Unique identifier for each programming record.
- **student_id (Foreign Key)**: References the `student_id` in the **Students Table**.
- **language**: Programming language being evaluated (e.g., Python, SQL).
- **problems_solved**: Total number of coding problems solved by the student.
- **assessments_completed**: Number of assessments completed by the student.
- **mini_projects**: Number of mini projects submitted by the student.
- **certifications_earned**: Number of programming certifications earned by the student.
- **latest_project_score**: Score received in the most recent programming project.

Soft Skills Table

This table stores data on students' performance in soft skills evaluations.

- **soft_skill_id (Primary Key)**: Unique identifier for each soft skill record.
- **student_id (Foreign Key)**: References the `student_id` in the **Students Table**.
- **communication**: Communication skills score (out of 100).
- **teamwork**: Teamwork skills score (out of 100).
- **presentation**: Presentation skills score (out of 100).
- **leadership**: Leadership skills score (out of 100).
- **critical_thinking**: Critical thinking skills score (out of 100).
- **interpersonal_skills**: Interpersonal skills score (out of 100).

Placements Table

This table stores details related to students’ placement readiness and outcomes.

- **placement_id (Primary Key)**: Unique identifier for each placement record.
- **student_id (Foreign Key)**: References the `student_id` in the **Students Table**.
- **mock_interview_score**: Score in the mock interviews (out of 100).
- **internships_completed**: Total number of internships completed by the student.
- **placement_status**: Placement readiness status (e.g., Ready, Not Ready, Placed).
- **company_name**: Name of the company where the student got placed (if applicable).

- **placement_package**: Package offered during placement (in USD or local currency).
- **interview_rounds_cleared**: Number of interview rounds cleared by the student.
- **placement_date**: Date when the placement offer was received.

Data Set Explanation

The dataset simulates real-world student performance metrics for a data science course:

- Relationships: Tables are connected via `student_id`.
- Realism: Data is generated using **Faker** to mimic plausible student data.
- Preprocessing: Ensure data types align with schema. Validate relationships.

Project Deliverables

1. Python source code, following OOP principles.
2. Streamlit application code and hosted app (if possible).
3. SQL queries and insights in a separate file.
4. Documentation of project steps and insights.

Project Guidelines

1. **Coding Standards**: Follow PEP8 guidelines for Python.
2. **Version Control**: Use Git for tracking changes.
3. **Documentation**: Comment code and maintain README files.
4. **Best Practices**:
    - Use environment variables for database credentials.
    - Structure code modularly using classes and methods.

Timeline:

2 weeks

References:

Project Live Evaluation	Project Live Evaluation
EDA Guide	Exploratory Data Analysis (EDA) Guide
Capstone Explanation Guideline	Capstone Explanation Guideline
GitHub Reference	How to Use GitHub.pptx
Project Orientation (English)	
Project Orientation (Tamil)	Recording Link
Project Excellence Series (English)	Python and SQL
Project Excellence Series (Tamil)	Python and SQL

REFERENCES:

- **Streamlit - Recording(Recording) (TAMIL)**
- **Special session for STREAMLIT(11/08/2024) (ENGLISH)**
- **STREAMLIT DOCUMENTATION:https://docs.streamlit.io/get-started/installation**

PROJECT DOUBT CLARIFICATION SESSION (PROJECT AND CLASS DOUBTS)

About Session: The Project Doubt Clarification Session is a helpful resource for resolving questions and concerns about projects and class topics. It provides support in understanding project requirements, addressing code issues, and clarifying class concepts. The session aims to enhance comprehension and provide guidance to overcome challenges effectively.
Note: Book the slot at least before 12:00 Pm on the same day

Timing: Monday-Saturday (4:00PM to 5:00PM)

Booking link :https://forms.gle/XC553oSbMJ2Gcfug9

LIVE EVALUATION SESSION (CAPSTONE AND FINAL PROJECT)

About Session: The Live Evaluation Session for Capstone and Final Projects allows participants to showcase their projects and receive real-time feedback for improvement. It assesses project quality and provides an opportunity for discussion and evaluation.
Note: This form will Open only on Saturday (after 2 PM) and Sunday on Every Week

Timing:

For BADM and DE
Monday-Saturday (11:30AM to 1:00PM)

For DS and AIML
Monday-Saturday (05:30PM to 07:00PM)

Booking link : https://forms.gle/1m2Gsro41fLtZurRA

Created By:	Verified By:	Approved By:
Aravinth Meganathan	Shadiya P P	Nehlath Harmain