📚 Placement Eligibility Dashboard Documentation

✅ 1. Overview

This project implements a **Placement Eligibility Dashboard** with:

- **Students** table
- **Programming** table
- **SoftSkills** table
- **Placements** table
- Python classes for OOP data insertion
- A Streamlit app for dynamic & individual queries.

✅ 2. Database Manager

`` uses **kwargs to reuse DB config:

```
DB_CONFIG = {
    'host': 'localhost',
    'user': 'root',
    'password': '12345',
    'database': 'placement_db'
}
```

Usage:

```
db = Databasemanager(**DB_CONFIG)
```

✅ 3. Table Structure

- **Students**: Basic student info.
- **Programs**: Problems solved in Python & SQL, mini projects, scores.
- **SoftSkills**: Communication, teamwork, presentation, leadership, etc.
- **Placements**: Placement readiness & outcome data.

Each table is created with foreign keys referencing `Students`.

✅ 4. Classes (Models)

Each table has a class with `save_to_db()` method:

- `Student`
- `Programming`
- `SoftSkill`
- `Placement`

They insert rows into respective tables.

✓ 5. Faker Data Generation

- 500 students generated using `Faker`.
- Random realistic data for programs, soft skills & placements.
- Placement records link to status & company.

✓ 6. Streamlit Dashboard

- **Dynamic filter**: Multiple inputs → combined filter on problems solved, softskills, mocks, package.
- **10 solo queries**: E.g., top packages, students placed, not placed, company wise count, min Python, min SQL, min softskill, teamwork, communication, mock interview.
- Uses `selectbox` for user choice.
- Runs appropriate SQL joins.

✓ 7. How to Run

1 Create DB: Run `.ipynb` blocks step-by-step.

2 Insert fake data: Run Faker loops.

3 Start Streamlit:

```
streamlit run placement_dashboard.py
```

4 Use sidebar to choose query & see results.

This covers your **GUVI placement_dashboard** as described.
