Internship Placement Manager: Project Charter

1.0 Project Overview

Project Name	**Internship Placement Manager**
Project Sponsor	School Administration
Project Manager	Project Team

1.1 Purpose and Justification

The primary purpose of this project is to develop a robust backend system that centralizes and streamlines the management of student internship placements. The current manual process of tracking students, employers, and placement details is inefficient and prone to errors. This new system will provide a single source of truth, improve data accuracy, and enable automated management of the entire internship lifecycle.

2.0 Project Objectives and Scope

2.1 Project Objectives

- To create a scalable, secure, and well-documented API for managing student, employer, mentor, and placement data.
- To provide an efficient tool for the Internship Coordinator (Sailor Moon) to track student progress and manage placement assignments.
- To establish a centralized database that can be integrated with future front-end applications.

- To ensure the system is easy to deploy and maintain using Docker containers.

2.2 Scope

- **In-Scope:**
    - Development of a RESTful API using **FastAPI** to manage all core entities (students, employers, mentors, placements, evaluations).
    - Implementation of a **PostgreSQL** database to persist all data.
    - Containerization of the application and database using **Docker** and **Docker Compose**.
    - Creation of comprehensive API documentation.
- **Out-of-Scope:**
    - Development of a front-end user interface.
    - Advanced features such as real-time notifications, chat functionality, or student-facing dashboards.
    - Complex reporting or analytics tools beyond basic data retrieval.

3.0 Key Deliverables and Success Criteria

3.1 Key Deliverables

- A fully functional, containerized backend API.
- A complete **PostgreSQL** database schema.
- Executable API documentation accessible at /docs.
- A **docker-compose.yml** file for easy deployment.
- Source code repository with a defined branch management strategy (main, dev, feature).

3.2 Success Criteria

- The API successfully performs all CRUD (Create, Read, Update, Delete) operations as defined in the test plan.
- All endpoints return the correct status codes and data formats.
- The application can be deployed on a local machine with a single command (docker-compose up --build).

4.0 Stakeholders and High-Level Timeline

4.1 Key Stakeholders

- **Project Sponsor:** Provides funding and strategic direction.
- **Internship Coordinator (Sailor Moon):** The primary end-user who will manage placements.
- **Students (Naruto):** The beneficiaries of the system, whose data will be managed.
- **Recruiters (Luffy):** Employers who will use the system to post internships.
- **Project Team:** Responsible for the design, development, and deployment of the application.

4.2 High-Level Timeline

- **Phase 1: Planning & Design (Week 1):** Define requirements, architecture, and database schema.
- **Phase 2: Core Development (Weeks 2):** Build the API endpoints and database logic.
- **Phase 3: Testing & Deployment (Week 3):** Conduct comprehensive testing and prepare for deployment.