

# Internship Placement Manager: Project Report

## 1.0 Project Overview

The Internship Placement Manager is a backend app designed to help schools manage internship placements. It tracks students, employers, mentors, and internship details. Built with **FastAPI** and **PostgreSQL**, the system runs in **Docker** containers for easy setup and deployment.

## 2.0 User Manual: Getting Started

### 2.1 Requirements

- **Docker:** Handles all dependencies and runs the app/database.
- **Web browser or API client:** Use for accessing docs or testing endpoints (e.g., Postman).

### 2.2 How to Run the App

1. **Navigate to your project folder** (where Dockerfile and docker-compose.yml are).
2. **Start the containers:**

```
docker-compose up --build
```

3. Check if it works:

Open <http://localhost:8000/docs> in your browser.

You should see the interactive API documentation.

### 3.0 Test Plan

#### Students Section

Test Case	What I'm Testing	How to Test It	Expected Result
Student-1	Add new student	POST /api/students/ with info	201 Created, new student details
Student-2	Add duplicate student	POST /api/students/ with existing email	Error: email already registered
Student-3	List all students	GET /api/students/	List of all students
Student-4	Get student by ID	GET /api/students/{stude nt_id}	Details for that student
Student-5	Get non-existent student	GET /api/students/{stude nt_id} (bad ID)	404 Not Found
Student-6	Update student info	PATCH /api/students/{stude nt_id}	Updated student info
Student-7	Delete student	DELETE	204 No Content,

		/api/students/{student_id}	then 404 on GET
--	--	----------------------------	-----------------

## Employers Section

Test Case	What I'm Testing	How to Test It	Expected Result
Employer-1	Add new employer	POST  /api/employers/	201 Created,  employer details
Employer-2	Add duplicate employer	POST  /api/employers/ with existing name	Error: company already registered
Employer-3	Get employer by ID	GET  /api/employers/{employer_id}	Employer details
Employer-4	Delete employer	DELETE  /api/employers/{employer_id}	204 No Content,  then 404 on GET

## Mentors Section

Test Case	What I'm Testing	How to Test It	Expected Result
-----------	------------------	----------------	-----------------

Mentor-1	Add new mentor	POST /api/mentors/	201 Created, mentor details
Mentor-2	Add duplicate mentor	POST /api/mentors/ with existing email	Error: email already registered
Mentor-3	Get mentor by ID	GET  /api/mentors/{mentor_id}	Mentor details

### Placements Section

Test Case	What I'm Testing	How to Test It	Expected Result
Placement-1	Create placement	POST  /api/placements/ with valid IDs	201 Created, placement details
Placement-2	Fake student	POST  /api/placements/ with bad student ID	404 Not Found
Placement-3	Fake employer	POST  /api/placements/ with bad employer ID	404 Not Found

Placement-4	List placements	GET  /api/placements/	List of placements
Placement-5	Get placement by  ID	GET  /api/placements/{placement_id}	Placement details

## 4.0 Personas & User Stories

### Personas

- **Naruto (Student):** Wants to update info, track applications, and see feedback.
- **Sailor Moon (Coordinator):** Manages placements, adds students/employers/mentors, matches them.
- **Luffy (Recruiter):** Posts internships, views student profiles.

### User Stories & Acceptance Criteria

#### Managing Student Information

- *As a student, I want to create/update my profile so the coordinator can see my info.*
  - System accepts full name, email, major.
  - Email must be unique.
  - Changes are confirmed.

#### Managing Companies and Internships

- *As a recruiter, I want to add my company and create internship listings.*
  - System accepts company name, industry, website.
  - Listings include job title, dates, company ID.

#### Matching Students and Internships

- *As a coordinator, I want to match students with employers and mentors.*

- System creates placements by linking student\_id, employer\_id, mentor\_id.
- Placement creation is confirmed.
- Can list all placements with details.

## Feedback and Evaluation

- *As a coordinator, I want to add evaluations for student internships.*
  - System accepts rating/comments, links to student and placement.
  - Students can view their evaluations.

## 5.0 Database Design and Plan

### Entity-Relationship Diagram (ERD)

- **Students:** Info about students (name, email, major).
- **Employers:** Info about companies.
- **Mentors:** Info about mentors.
- **Placements:** Links students, employers, mentors for each internship.
- **Evaluations:** Performance reviews for placements.

*Tables are linked by foreign keys (e.g., student\_id in Placements links to Students).*

### Normalization Notes

- No repeated data; use IDs and foreign keys.
- Easy updates and clean data.

### Migration Plan

- On first run, app creates all tables.
- If tables are empty, app adds sample data.

## 6.0 Architecture and Technical Stuff

### System Architecture

- **API Layer (src/api):** Defines endpoints.
- **Business Logic Layer (src/schemas.py, src/data\_models.py):** Data validation and structure.
- **Database Layer (src/connections.py):** Handles DB communication.

## Error Model

- Uses HTTPException for clear error messages and status codes.

## Authentication Strategy

- Students log in with email/password.
- App issues JWT token for authenticated requests.

## 7.0 Team & Project Management

### Repository Setup

- **main:** Production-ready code.
- **dev:** Feature integration and testing.
- **feature branches:** For new features/bug fixes.

### CI/CD Pipeline

- Runs tests and code checks on every push.
- Deploys if tests pass.

### Project Board

- **To Do:** Tasks to start.
- **In Progress:** Tasks being worked on.
- **In Review:** Finished tasks needing review.
- **Done:** Completed tasks.

**Made with FastAPI, PostgreSQL, SQLAlchemy, Alembic, and Docker**