# Mini Project Plan – Team Fostering Technical Test

## Project Title:

Student Grade Processing System

## Objective:

To build a modular, SQL-backed system that  
- Stores and manages student data (ID, name, DOB, module grades, etc.)  
- Calculates average grades and classifies academic performance  
- Provides REST API endpoints for external clients (web, AI)  
- Reflects scalable, real-world software practices aligned with the Dynamic Team Fostering (DTF) system

## Scope of Work:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Feature | Description | | MySQL Database | Stores structured student data using relational models | | Backend API | Laravel-powered REST API that handles CRUD operations and grading logic | | Frontend Client | Python Flask and HTML | | Grading Logic | Average calculation and classification: Pass (40%), Merit (60%), Distinction (70%) | | Report Export | Export student summaries to CSV | | (Optional) Mock AI Service | Simulated endpoint for future speech-to-text or NLP integration | | Ethical Consideration | Commentary on secure data handling, GDPR, and trauma-aware design | |

## Technology Stack

|  |  |
| --- | --- |
| Component | Tool/Language |
| Database | MySQL |
| Backend API | Laravel (PHP) |
| Frontend | Python (CLI or GUI using Tkinter) |
| Data Export | Python CSV/JSON export |
| Testing | Postman for API testing (optional) |
| Version Control | Git + GitHub |
| (Optional) | Docker for containerized development (e.g., Laravel + MySQL) |

## Timeline (4 Days Estimate):

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Day | Task | | Day 1 | Project Plan Development | | Day 2 | Project Architecture | | Day 3 | Backend and Frontend Development | | Day 4 | Github Deployment and Video Making | |

## Deliverables:

- Laravel project with MySQL-backed student API  
- Python frontend code to interact with the API  
- Sample SQL data for 10 students  
- README.md with setup instructions and architecture overview  
- GitHub repository with all code and documentation