

Project Plan

Project Title: Internship Management Portal

Team Number: 14

Team Members:

Name	Roll No	Mobile Number	Email
A. Prashanth	22071A0565	6305763482	prashanthagithe@gmail.com
CH. Akhil Sai Charan	22071A0571	9949555269	akhil639charan@gmail.com
V. Hemanth Kumar	22071A0579	9121059539	vrehemanth@gmail.com
M. Sai Varshith	22071A0592	7013373387	varshith2110@gmail.com
N. Vishnu Vardhan	22071A0596	7702828675	n.vishnuvardhan2004@gmail.com

1. Project Overview:

This project manages student records with CRUD operations, secure JWT-based APIs, and file uploads stored in Firebase. It features downloadable templates, monthly filtered reports, and a React frontend integrated with a Node.js backend and MongoDB database. Testing and deployment use modern tools ensuring reliability and scalability.

2. Requirement Analysis

Functional Requirements:

Student Record Management:

- CRUD operations for student records.
- Fields: Name, Roll No, Branch, Company Name, Duration, Stipend.

File Upload:

- Upload Offer Letter, NOC, Completion Certificate (PDF/DOCX).
- Store in Firebase and link to student record.

Downloadable Templates:

- API endpoints to download NOC and LOR templates (from pre-uploaded .docx/.pdf files).

Monthly Reports:

- Generate report (by branch, month, company).
- Export format: PDF or table.

Non-Functional Requirements:

- Secure APIs (JWT-based auth).
- Responsive performance (API latency < 1 sec).
- Role-based access (Admin, Faculty, Student – if applicable).

3. System Design

- **Frontend:** React.js with Axios for API calls, Firebase SDK for uploads.
- **Backend:** Node.js + Express.js (REST API).
- **Database:** MongoDB (student records, document URLs, metadata).
- **File Storage:** Firebase Storage.
- **Authentication:** JWT (for admin/faculty login).

4. Implementation Plan

Tech Stack:

- **Frontend:** React.js, React Router, Shadcn
- **Backend:** Node.js, Express.js
- **Database:** MongoDB (with Mongoose ODM)
- **File Management:** Firebase Storage
- **Authentication:** JWT

5. Testing

- Unit Testing: Jest (Node.js), React Testing Library (Frontend)
- Integration Testing: Postman / Insomnia for API testing
- Firebase Uploads: Manual + Programmatic checks

Test Cases:

- Valid/invalid document uploads
- Role-based API access
- Report generation with filters
- CRUD flows

6. Deployment

- Frontend: Vercel
- Backend: Render
- MongoDB: MongoDB Atlas
- Firebase: Google Firebase Console (Storage + Admin SDK)

7. Maintenance

- Monitor Firebase usage limits
- Regular database backups (monthly)

Work Division

Person	Role	Key Responsibilities	Tech Stack / Tools
A.Prashanth (22071A0565)	Backend Developer 1	- Design MongoDB schema- Implement CRUD APIs for student records- Generate filtered reports- Unit & integration testing	Node.js, Express.js, MongoDB, Jest, Postman
N.Vishnu Vardhan Reddy (22071A0596)	Backend Developer 2	- JWT-based authentication- Store Firebase file URLs in DB- API endpoints for NOC/LOR template downloads- Assist in backend testing	Node.js, Express.js, Firebase Admin SDK
M.Sai Varshith (22071A592)	Frontend Developer 1	- React app setup and routing- UI for Add/Edit Student forms with validation- Design Table/Grid views for records	React.js, React Router, CSS/Material UI
Ch.Akhil Sai Charan (22071A0571)	Frontend Developer 2	- Integrate REST APIs with Axios- File upload/download UI components- Report view UI with filters	React.js, Axios, CSS/Material UI
V.Hemanth Kumar (22071A0579)	Firebase Integration	- Setup Firebase Storage & rules- Implement upload logic for Offer Letter, NOC, Certificate functionality	Firebase SDK, Firebase Console