

Internship Management System - Project Proposal

1. Project Overview

The **Internship Management System (IMS)** is a web-based application developed using **Angular 20 (frontend)** and **.NET 8 Web API (backend)**. The system is designed to streamline the management of internships for training sessions. It provides features for both **Admin (Trainer)** and **Intern (Student)** roles with role-based access control. The application will be reusable for future training batches.

2. Objectives

- Provide a platform for trainers to manage students, batches, courses, attendance, progress, and assessments.
 - Provide a platform for interns to register (with profile photo upload), view their progress, submit queries (QnA with file support), and access their assessments and attendance records.
 - Implement a **QnA system** where all questions and answers (with attachments like PDF/Excel) are visible to every intern, ensuring collective learning.
 - Enable Admins to track performance and attendance through a dedicated **dashboard with graphs**.
 - Automate the assignment of the **Intern role** for new registrations, while Admin credentials are pre-seeded.
-

3. Roles & Access

Admin (Trainer)

- Pre-seeded account (login only).
- Manage Users (CRUD).
- Manage Batches, Courses, and Internships.
- Approve/Reject Internship Applications.
- Track Intern Progress & Attendance.
- Assign Assessments and Scores.
- Answer Intern QnAs.
- Generate Reports (PDF/Excel).
- Dashboard with charts showing attendance & progress.

Intern (Student)

- Register/Login (automatically assigned as Intern) with **photo upload**.
- View their assigned batch and courses.
- Apply for internship programs.
- Submit daily progress reports.
- View their attendance and assessment results.

- Participate in QnA (ask/view questions, attach PDF/Excel, view answers).

4. Core Modules (Models)

1. **User** – Id, Name, Email, PasswordHash, Role (Admin/Intern), **ProfilePhotoPath**
2. **Batch** – Id, Name, StartDate, EndDate
3. **UserBatch** – Id, UserId, BatchId (links Users and Batches)
4. **Course** – Id, Title, Description, Duration, BatchId
5. **Attendance** – Id, UserId, Date, Status (Present/Absent)
6. **Assessment** – Id, CourseId, Title, MaxScore
7. **AssessmentResult** – Id, AssessmentId, UserId, Score
8. **DailyProgress** – Id, UserId, Date, TaskDone, Status
9. **QnA** – Id, UserId, Question, Answer, FileAttachmentPath, CreatedAt
10. **InternshipProgram** – Id, Title, Description, Duration, MentorName
11. **Application** – Id, UserId, InternshipId, Status (Pending/Approved/Rejected)

5. File Storage Strategy

- **Profile Photos:** Stored in **server file system** by default (e.g., `/uploads/profilephotos/` inside backend). The database saves only the file path (`ProfilePhotoPath`). For production-ready deployments, this can be extended to **cloud storage** (Azure Blob / AWS S3), which offers better scalability and reliability.
- **QnA Attachments:** Interns can upload **PDF/Excel files**. By default, these are stored in the **server file system** (e.g., `/uploads/qna/`). The database saves the reference path (`FileAttachmentPath`). For advanced setups, files can alternatively be stored in **cloud storage** for persistence and CDN delivery.

6. Role-Based Access Matrix

Model	Admin Access	Intern Access
User	✓ CRUD	✗
Batch	✓ CRUD	✗
UserBatch	✓ CRUD	View Only
Course	✓ CRUD	✗
Attendance	✓ CRUD	View Own
Assessment	✓ CRUD	View Only

Model	Admin Access	Intern Access
AssessmentResult	✓ CRUD	View Own
DailyProgress	✓ CRUD	✓ CRUD Own
QnA	✓ CRUD	✓ Ask/View + File Upload
Internship	✓ CRUD	Apply/View
Applications	✓ Approve	✓ Apply

7. Dashboards

Admin Dashboard

- Attendance percentage per intern.
- Progress tracking (daily progress completion).
- Batch-wise performance statistics.

Intern Dashboard

- Daily progress tracker.
- Attendance record.
- Assessment results.
- Internship application status.
- QnA board with file support.

8. API Endpoints (Illustrative)

Auth

- `POST /api/auth/login` – Admin & Intern login.
- `POST /api/auth/register` – Intern registration with **profile photo upload**.

Users

- `GET /api/users` – Admin only.
- `POST /api/users` – Admin only.
- `PUT /api/users/:id` – Admin only.
- `DELETE /api/users/:id` – Admin only.

Batches & UserBatch

- `POST /api/batches` – Admin.
- `GET /api/batches` – All.

- POST /api/userbatch – Admin.
- GET /api/userbatch/:userId – Intern (own only).

Attendance

- POST /api/attendance – Admin.
- GET /api/attendance/:userId – Intern (own).
- GET /api/attendance – Admin (all).

Assessments

- POST /api/assessments – Admin.
- GET /api/assessments – Intern (view only).
- POST /api/assessments/:id/result – Admin.
- GET /api/assessments/:id/result/:userId – Intern (own).

Daily Progress

- POST /api/dailyprogress – Intern.
- GET /api/dailyprogress/:userId – Intern (own), Admin (all).

QnA

- POST /api/qna – Intern (ask with optional PDF/Excel upload).
- PUT /api/qna/:id – Admin (answer).
- GET /api/qna – All users.

Internship Applications

- GET /api/internships – All.
- POST /api/internships – Admin.
- POST /api/applications – Intern.
- PUT /api/applications/:id – Admin.
- GET /api/applications/:userId – Intern (own).

9. Frontend Structure (Angular)

```

/app
  /auth          -> login, register (with photo upload)
  /admin
    /dashboard    -> charts, analytics
    /users        -> CRUD
    /batches      -> CRUD
    /courses      -> CRUD
    /attendance   -> manage/view
    /assessments  -> CRUD + scores

```

```
/internships -> CRUD + applications
/qna          -> manage all QnA (with file download)
/intern
  /dashboard  -> personal progress, attendance
  /qna        -> ask/view (with file upload)
  /progress   -> submit/view progress
  /applications -> apply internship
/shared
  /models
  /services
  /guards
```

10. Development Roadmap

Phase 1 – Setup

- Angular 20 app with routing & guards.
- Backend setup (.NET 8 Web API + SQL Server).
- JWT Authentication with seeded Admin account.

Phase 2 – Core Models

- Implement Users (with profile photo), Batches, UserBatch, Courses.
- Auto-assign Intern role at registration.

Phase 3 – Intern Features

- QnA board (with file upload).
- Daily Progress submission.
- Attendance & Assessment view.
- Internship Applications.

Phase 4 – Admin Features

- Manage all models (CRUD).
- Approve/Reject applications.
- Assign scores & attendance.

Phase 5 – Dashboards & Reports

- Charts (attendance, progress).
 - Export reports (PDF/Excel).
 - Deployment (Docker, Azure, or Netlify/Vercel).
-

11. Conclusion

This **Internship Management System** will serve as a comprehensive solution for managing interns during training sessions. With role-based access, dashboards, collaborative QnA (with PDF/Excel support), and profile photo upload during registration, it ensures streamlined operations for trainers while enhancing the learning experience for interns.