

Data Flow of the Student Admission + Document Management System

1. Student Registration & Login

Frontend (Angular)

- Student fills Register Form → POST /api/studentaccount/register.
- Login → POST /api/studentaccount/login.

Backend (ASP.NET Core API)

- Saves student inAspNetUsers (Identity) table.
- Student details stored in Student Table (linked by UserId).
- On login → API validates credentials → returns JWT token (or session details).

Database

- AspNetUsers → authentication (email, password).
- Students → personal + academic info (Name, DOB, Marks, CourseApplied).

2. Student Admission Application

Frontend

- Student fills admission form → POST /api/admission/apply.
- Can view application status → GET /api/admission/status/{studentId}.

Backend

- Saves admission details in Admissions table.
- Default status = Pending.

Database

- Admissions → StudentId, CourseId, Status (Pending/Approved/Rejected).
- Courses → list of available courses.

3. Student Uploads Documents

Frontend

- Student uploads PDF → POST /api/documents/upload.

Backend

- Validates file → saves in /uploads/studentId/yyyy/MM/.
- Stores path + metadata in Documents table.

Database

- Documents → DocumentId, StudentId, FilePath, Type, Status.

4. Admin Login

Frontend

- Admin login form → POST /api/account/login.

Backend

- Checks role (Admin/User).
- Issues token for Admin.

Database

- AspNetUsers with Role = Admin.

5. Admin Reviews Students

Frontend

- Admin dashboard → GET /api/students/all.
- Shows list with filter/search.

Backend

- Fetches all students from DB with admission + document details.

Database

- Students + Admissions + Documents join query.

6. Admin Approves / Rejects

Frontend

- Admin clicks Approve/Reject → PUT /api/admission/updateStatus/{studentId}.

Backend

- Updates Admissions.Status field.

Database

- Status updated in Admissions.

7. Admin / Student Document Access

Frontend

- Admin sees uploaded documents list → GET /api/documents/student/{studentId}.

- Download document → GET /api/documents/download/{docId}.

Backend

- Reads path from DB → streams file.

Database

- Documents → used only for metadata (actual file in server/cloud).

Overall Data Flow Diagram (Textual)

(Student Angular App)

|
| Register/Login (POST /studentaccount/*)
v

(ASP.NET Core API)

|
| Save user -> AspNetUsers
| Save student info -> Students
v

(Database)

(Student Angular App)

|
| Apply Admission (POST /admission/apply)
v

(ASP.NET Core API)

|
| Save admission -> Admissions
v

(Database)

(Student Angular App)

|
| Upload Document (POST /documents/upload)
v

(ASP.NET Core API)

- |
- | Save file -> /uploads/studentId/yyyy/MM/
- | Save path -> Documents
- v

(Database + File System)

(Admin Angular App)

- |
- | Login (POST /account/login) -> Role=Admin
- | View students (GET /students/all)
- | View documents (GET /documents/student/{id})
- | Approve/Reject (PUT /admission/updateStatus)
- v

(ASP.NET Core API)

- |
- | Reads/Updates Students, Admissions, Documents
- v

(Database + File System)