

API Documentation - Attendance Tracking System

Authentication

POST /api/auth/login

- Description: Logs in an admin user.

- Request Body:

```
{
  "email": "admin@example.com",
  "password": "yourPassword"
}
```

- Response:

```
{
  "message": "Login successful",
  "user": {
    "id": "adminId",
    "fullName": "Admin Name",
    "email": "admin@example.com",
    "university": "University Name"
  }
}
```

- Note: A JWT cookie is returned for authentication.

Lecturers

POST /api/lecturer/create [Protected]

- Description: Add a new lecturer.

- Request Body: JSON object with lecturer details.

- Response: { "id": "generatedLecturerId" }

GET /api/lecturer [Protected]

- Description: Get list of all lecturers.

- Response: Array of lecturer objects.

GET /api/lecturer/:id

- Description: Get details of a specific lecturer by ID.

- Response: Lecturer object or 404 if not found.

Departments

POST /api/department/create [Protected]

- Description: Create a new department.

- Request Body: JSON object with department info.

- Response: { "id": "generatedDepartmentId" }

GET /api/department [Protected]

- Description: List all departments.

- Response: Array of department objects.

API Documentation - Attendance Tracking System

PUT /api/department/:id [Protected]

- Description: Update department by ID.
- Request Body: Updated fields.
- Response: { "message": "Department updated successfully" }

DELETE /api/department/:id [Protected]

- Description: Delete department by ID.
- Response: { "message": "Department deleted successfully" }

Dashboard

GET /api/dashboard/overview [Protected]

- Description: Returns dashboard data.
- Response:

```
{
  "totalStudents": 50,
  "totalClasses": 20,
  "attendancePercentage": 85,
  "absentRate": 15
}
```

Attendance

GET /api/attendance [Protected]

- Description: Get all students' attendance summary.
- Response: Array of student attendance objects.

GET /api/attendance/:regNumber [Protected]

- Description: Get a specific student's profile info.
- Response: Object with student's name, dept, ID, phone, email.

Authentication Middleware

- Middleware: protectRoute
- Description: Checks for a valid JWT token in cookies.
- If not present or invalid, returns 401 or 403.
- Usage: Applied on protected routes to restrict access to authenticated admins.