ChatGPT

Fred Attendance System – API Documentation

Base URL: `http://<your-domain>:7000/api`

Authentication

POST `/auth/login`

Authenticate an admin user and return a JWT cookie.

Request Body:

```
{
  "email": "admin@example.com",
  "password": "admin123"
}
```

Response:

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

GET `/auth/logout`

Logs the admin out by clearing the JWT cookie.

Response:

```
{
  "message": "Logged out successfully"
}
```

Lecturers

POST `/lecturer/create`

Create a new lecturer.

Request Body:

```
{
  "fullname": "Dr. John Smith",
  "email address": "john@uni.edu",
  "department": "Computer Science",
  "phone number": "08012345678",
  "noOfCourses": 2
}
```

Response:

```
{
  "id": "lecturer_doc_id"
}
```

GET `/lecturer/`

Get all lecturers.

Response:

```
[
  {
    "id": "lecturerId",
    "fullname": "Dr. John Smith",
    "department": "Computer Science",
    ...
```

```
    }
  ]
```

GET `/lecturer/:id`

Get a lecturer by ID.

Response:

```
{
  "id": "lecturerId",
  "fullname": "Dr. John Smith",
  ...
}
```

Departments

POST `/department/create`

Create a new department.

Request Body:

```
{
  "name": "Computer Science",
  "school": "Physical Sciences"
}
```

GET `/department/`

Get all departments.

Response:

```
[
  {
    "id": "departmentId",
    "name": "Computer Science",
    "school": "Physical Sciences"
```

```
    }
  ]
```

PUT `/department/:id`

Update a department by ID.

Request Body:

```
{
  "name": "New Dept Name"
}
```

DELETE `/department/:id`

Delete a department by ID.

📊 Dashboard

GET `/dashboard/overview`

Get stats about students, classes, and attendance.

Response:

```
{
  "totalStudents": 300,
  "totalClasses": 50,
  "attendancePercentage": 82,
  "absentRate": 18
}
```

📈 Attendance

GET `/attendance/`

Get attendance rates of all students.

Response:

```json
[
  {
    "name": "Jane Doe",
    "studentID": "2020CS1234",
    "email": "jane@example.com",
    "attendanceRate": "85%",
    "classesAttended": "17/20"
  }
]
```

GET `/attendance/:regNumber`

Get student profile and basic attendance info by registration number.

Example: `/attendance/2020CS1234`

Response:

```json
{
  "name": "Jane Doe",
  "department": "Computer Science",
  "studentID": "2020CS1234",
  "phone": "08012345678",
  "email": "jane@example.com"
}
```

Firestore Structure

```
- students
  - <regNumber>
- courses
  - <courseId>
    - students
      - <regNumber>
    - attendance
      - <attendanceDocId>
- lecturer
- department
- admins
```

```
- classes
- attendance
```