

# API Documentation

## 1 Register User

**Endpoint:** POST /register

Registers a new user.

**Request Body:**

```
{
  "auth0_id": "string",
  "username": "string",
  "name": "string"
}
```

**Responses:**

- 200 OK: User registered successfully.
- 400 Bad Request: Invalid request body.
- 401 Unauthorized: Invalid token.
- 500 Internal Server Error: Database error.

## 2 Login User

**Endpoint:** POST /login

Logs in an existing user.

**Request Body:**

```
{
  "auth0_id": "string"
}
```

**Responses:**

- 200 OK: Login successful.
- 400 Bad Request: Invalid request body.
- 404 Not Found: User not found.

## 3 Delete User

**Endpoint:** DELETE /delete

Deletes an existing user.

**Headers:**

Authorization: Bearer <token>

**Responses:**

- 200 OK: User deleted successfully.
- 401 Unauthorized: Invalid token.
- 500 Internal Server Error: Database error.

## 4 Get User Courses

**Endpoint:** GET /courses

Retrieves the courses of the authenticated user.

**Headers:**

Authorization: Bearer <token>

**Responses:**

- 200 OK: List of courses.
- 401 Unauthorized: Invalid token.
- 500 Internal Server Error: Database error.

## 5 Join Course

**Endpoint:** POST /join

Joins a course for the authenticated user.

**Request Body:**

```
{  
  "course_name": "string"  
}
```

**Headers:**

Authorization: Bearer <token>

**Responses:**

- 200 OK: Course joined successfully.
- 400 Bad Request: Invalid request body.
- 401 Unauthorized: Invalid token.
- 500 Internal Server Error: Database error.

## 6 Leave Course

**Endpoint:** POST /leave

Leaves a course for the authenticated user.

**Request Body:**

```
{  
  "course_name": "string"  
}
```

**Headers:**

Authorization: Bearer <token>

**Responses:**

- 200 OK: Course left successfully.
- 400 Bad Request: Invalid request body.
- 401 Unauthorized: Invalid token.
- 500 Internal Server Error: Database error.

## 7 Create Message

**Endpoint:** POST /chats/:chatId/messages

Creates a message in a specified chat.

**Request Body:**

```
{  
  "text": "string"  
}
```

**Headers:**

Authorization: Bearer <token>

**Responses:**

- 201 Created: Message created successfully.
- 400 Bad Request: Invalid request body.
- 404 Not Found: Chat or user not found.
- 500 Internal Server Error: Internal error.

## 8 Get Messages

**Endpoint:** GET /chats/:chatId/messages

Retrieves messages from a specified chat.

**Responses:**

- 200 OK: List of messages.
- 404 Not Found: Chat or messages not found.

## 9 Get Course

**Endpoint:** POST /course

Retrieves details of a specific course.

**Request Body:**

```
{  
  "course_name": "string"  
}
```

**Responses:**

- 200 OK: Course details.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 10 Create Course

**Endpoint:** POST /createCourse

Creates a new course.

**Request Body:**

```
{  
  "course_name": "string"  
}
```

**Responses:**

- 200 OK: Course created successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 11 Delete Course

**Endpoint:** DELETE /deleteCourse

Deletes a specific course.

**Request Body:**

```
{  
  "course_name": "string"  
}
```

**Responses:**

- 200 OK: Course deleted successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 12 Get All Courses

**Endpoint:** GET /allCourses

Retrieves a list of all courses.

**Responses:**

- 200 OK: List of all courses.
- 500 Internal Server Error: Database error.

## 13 Get Students

**Endpoint:** POST /students

Retrieves a list of students in a specific course.

**Request Body:**

```
{  
  "course_name": "string"  
}
```

**Responses:**

- 200 OK: List of students.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 14 Add Student

**Endpoint:** POST /addStudent

Adds a student to a specific course.

**Request Body:**

```
{  
  "course_name": "string",  
  "student_id": "string"  
}
```

**Responses:**

- 200 OK: Student added successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 15 Remove Student

**Endpoint:** POST /removeStudent

Removes a student from a specific course.

**Request Body:**

```
{
  "course_name": "string",
  "student_id": "string"
}
```

**Responses:**

- 200 OK: Student removed successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 16 Get Chat

**Endpoint:** POST /chat

Retrieves details of a specific chat.

**Request Body:**

```
{
  "chat_id": "string"
}
```

**Responses:**

- 200 OK: Chat details.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 17 Create Chat

**Endpoint:** POST /createChat

Creates a new chat.

**Request Body:**

```
{
  "chat_name": "string",
  "owner_id": "string",
  "additional_users": ["string"]
}
```

**Responses:**

- 200 OK: Chat created successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 18 Update Chat

**Endpoint:** POST /updateChat

Updates details of a specific chat.

**Request Body:**

```
{
  "chat_id": "string",
  "name": "string"
}
```

**Responses:**

- 200 OK: Chat updated successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 19 Delete Chat

**Endpoint:** DELETE /deleteChat

Deletes a specific chat.

**Request Body:**

```
{
  "chat_id": "string",
  "user_id": "string"
}
```

**Responses:**

- 200 OK: Chat deleted successfully.
- 400 Bad Request: Invalid request body.
- 401 Unauthorized: User is not the owner.
- 500 Internal Server Error: Database error.

## 20 Get All Chats

**Endpoint:** POST /allChats

Retrieves a list of all chats for a specific user.

**Request Body:**

```
{
  "user_id": "string"
}
```

**Responses:**

- 200 OK: List of all chats.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 21 Add User to Chat

**Endpoint:** POST /addUser

Adds a user to a specific chat.

**Request Body:**

```
{  
  "user_id": "string",  
  "chat_id": "string"  
}
```

**Responses:**

- 200 OK: User added successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 22 Remove User from Chat

**Endpoint:** POST /removeUser

Removes a user from a specific chat.

**Request Body:**

```
{  
  "user_id": "string",  
  "chat_id": "string"  
}
```

**Responses:**

- 200 OK: User removed successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 23 Get Room

**Endpoint:** POST /room

Retrieves details of a specific room.

**Request Body:**

```
{  
  "room_name": "string",  
  "building_name": "string"  
}
```

**Responses:**

- 200 OK: Room details.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 24 Create Room

**Endpoint:** POST /createRoom

Creates a new room.

**Request Body:**

```
{  
  "room_name": "string",  
  "building_name": "string"  
}
```

**Responses:**

- 200 OK: Room created successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 25 Delete Room

**Endpoint:** DELETE /deleteRoom

Deletes a specific room.

**Request Body:**

```
{  
  "room_name": "string",  
  "building_name": "string"  
}
```

**Responses:**

- 200 OK: Room deleted successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 26 Get Room Courses

**Endpoint:** POST /roomCourses

Retrieves a list of courses in a specific room.

**Request Body:**

```
{  
  "room_name": "string",  
  "building_name": "string"  
}
```

**Responses:**

- 200 OK: List of courses.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.



## 27 Add Course to Room

**Endpoint:** POST /addCourse

Adds a course to a specific room.

**Request Body:**

```
{
  "room_name": "string",
  "building_name": "string",
  "course_name": "string"
}
```

**Responses:**

- 200 OK: Course added successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 28 Remove Course from Room

**Endpoint:** POST /removeCourse

Removes a course from a specific room.

**Request Body:**

```
{
  "room_name": "string",
  "building_name": "string",
  "course_name": "string"
}
```

**Responses:**

- 200 OK: Course removed successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 29 Set Room Capacity

**Endpoint:** POST /setCapacity

Sets the capacity of a specific room.

**Request Body:**

```
{
  "room_name": "string",
  "building_name": "string",
  "capacity": "integer"
}
```

**Responses:**

- 200 OK: Capacity set successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 30 Set Room Occupancy

**Endpoint:** POST /setOccupancy

Sets the occupancy of a specific room.

**Request Body:**

```
{
  "room_name": "string",
  "building_name": "string",
  "occupancy": "integer"
}
```

**Responses:**

- 200 OK: Occupancy set successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 31 Add Occupant to Room

**Endpoint:** POST /addOccupant

Adds an occupant to a specific room.

**Request Body:**

```
{
  "room_name": "string",
  "building_name": "string",
  "student_id": "string"
}
```

**Responses:**

- 200 OK: Occupant added successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 32 Remove Occupant from Room

**Endpoint:** POST /removeOccupant

Removes an occupant from a specific room.

**Request Body:**

```
{
  "room_name": "string",
  "building_name": "string",
  "student_id": "string"
}
```

**Responses:**

- 200 OK: Occupant removed successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 33 Get Room Occupants

**Endpoint:** POST /occupants

Retrieves a list of occupants in a specific room.

**Request Body:**

```
{
  "room_name": "string",
  "building_name": "string"
}
```

**Responses:**

- 200 OK: List of occupants.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 34 Get Building

**Endpoint:** POST /building

Retrieves details of a specific building.

**Request Body:**

```
{
  "building_name": "string"
}
```

**Responses:**

- 200 OK: Building details.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 35 Create Building

**Endpoint:** POST /createBuilding

Creates a new building.

**Request Body:**

```
{
  "building_name": "string"
}
```

**Responses:**

- 200 OK: Building created successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 36 Delete Building

**Endpoint:** DELETE /deleteBuilding

Deletes a specific building.

**Request Body:**

```
{
  "building_name": "string"
}
```

**Responses:**

- 200 OK: Building deleted successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 37 Get Building Rooms

**Endpoint:** POST /buildingRooms

Retrieves a list of rooms in a specific building.

**Request Body:**

```
{  
  "building_name": "string"  
}
```

**Responses:**

- 200 OK: List of rooms.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 38 Add Room to Building

**Endpoint:** POST /addRoom

Adds a room to a specific building.

**Request Body:**

```
{  
  "building_name": "string",  
  "room_number": "string"  
}
```

**Responses:**

- 200 OK: Room added successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.

## 39 Remove Room from Building

**Endpoint:** POST /removeRoom

Removes a room from a specific building.

**Request Body:**

```
{  
  "building_name": "string",  
  "room_number": "string"  
}
```

**Responses:**

- 200 OK: Room removed successfully.
- 400 Bad Request: Invalid request body.
- 500 Internal Server Error: Database error.