

# Assignment Submission Portal

## Backend Design and Implementation

### 1. Introduction:

This is a backend system for an assignment submission portal. It allows users to upload assignments and admins to accept or reject them. The portal uses MongoDB for data storage and follows a modular and RESTful API design.

### 2. API Endpoints

#### 2.1 User End Points

##### 2.1.1 User Registrations

**Endpoint : POST/users/register/**

#It registers new user in the system

**Json:**

```
{
  "username": "your_username",
  "password": "your_password"
}
```

**Response:**

For success:

```
{
  "message": "User successfully registered"
}
```

For failure:

```
{
  "error": "User already Exist"
}
```

#### 2.2 User Login:

**Endpoint: POST/users/login/**

#logs the existing user in the system

**Json:**

```
{
```

```
    "username": "your_username",
    "password": "your_password"
  }
```

Response:

For success:

```
{
  "message": "Login Successfully"
}
```

For failure:

```
{
  "error": "Invalid credentials"
}
```

### 2.3 Upload Assignment:

Endpoints: POST/users/upload/

#Allows a user to upload an assignment tagged to a specific admin.

Json:

```
{
  "User_id" : "Your_Username"
  "task" : "Your_assignment_task"
  "admin" : "admin_username"
}
```

Response:

For success:

```
{
  "message": "Assignment uploaded successfully"
}
```

For failure:

```
{
  "error": "Admin not found"
}
```

### 2.4 Fetch Admins:

Endpoints: GET/users/admins/

#Allows a user to get all admins.

Json:

```
{
  "admins": [
    {"username": "admin1"},
    {"username": "admin2"}
  ]
}
```

Response:

```
{
  "error": "An unexpected error occurred"
}
```

### 3 Admin Endpoints:

#### 3.1 Admin Registrations:

Endpoints: POST/admins/register/

#Registers a new admin in the system.

Json:

```
{
  "username": "your_admin_username",
  "password": "your_admin_password"
}
```

Response:

For Success:

```
{
  "message": "Admin registered successfully"
}
```

For failure:

```
{
  "error": "Admin already exists"
}
```

#### 3.2 Admin Logins:

Endpoints: POST/admins/login/

#logins the admin

Json:

```
{
  "username": "your_admin_username",
  "password": "your_admin_password"
}
```

Response:

For success:

```
{
  "message": "Login successful"
}
```

For failure:

```
{
  "error": "Invalid credentials"
}
```

## 2.3 View Assignments

Endpoint: GET /admins/assignments/  
#able to view all assignments

Url: /admins/assignments/?admin=admin\_username

Response:

For success:

```
{
  "assignments": [
    {
      "_id": "645aef23599e7b21f8e2dabc",
      "userId": "john_doe",
      "task": "Complete the backend API design",
      "admin": "alok_admin",
      "status": "pending",
      "timestamp": "2024-11-17T14:30:00"
    }
  ]
}
```

For failure:

```
{
  "error": "No assignments found for the specified admin."
}
```

## 2.4 Accept an Assignment:

**Endpoint:** POST /admins/assignments/<assignment\_id>/accept/

Response:

For success:

```
{
  "message": "Assignment accepted successfully"
}
```

For failure:

```
{
  "error": "Assignment not found"
}
```

## 2.5 Reject an Assignment:

Endpoint: POST /admins/assignments/<assignment\_id>/reject/

Response:

For success:

```
{
```

```
    "message": "Assignment rejected successfully"
  }
```

For failure:

```
{
  "error": "Assignment not found"
}
```

#### 4 Error Handling:

Invalid Assignment:

```
{
  "error": "Invalid assignment ID format"
}
```

Assignment Not Found:

```
{
  "error": "Assignment not found"
}
```

Database Connection Issue:

```
{
  "error": "Could not connect to the database"
}
```

Invalid HTTP Method:

```
{
  "error": "Only POST requests are allowed."
}
```

#### 5 Future Enhancements:

- ✧ Add token-based authentication for enhanced security.
- ✧ Add pagination for GET /admins/assignments/ for better performance with large datasets.
- ✧ Include email or notification systems for assignment status updates.