

Initialize the Project

```
md or mkdir usermanagement
```

```
cd usermanagement
```

```
npm init -y
```

Install dependencies

```
npm install express mongoose bcryptjs jsonwebtoken body-parser
```

Push files/folders to Github

```
echo "# backendassgn" >> README.md
```

```
git init
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git branch -M main
```

```
git remote add origin https://github.com/codehelm/backendassgn.git
```

```
git push -u origin main
```

Create the Database Schema

```
models/User.js
```

```
models/Document.js
```

Create Routes and Controllers

```
routes/userRoutes.js
```

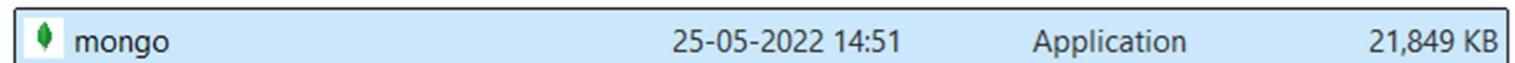
```
routes/adminRoutes.js
```

Set up the Server

```
index.js
```

Run the application

Invoke this application



Run the Server

```
PS C:\user\usermanagement> node index.js
(node:20716) [MONGODB DRIVER] Warning: use
be removed in the next major version
(Use `node --trace-warnings ...` to show w
(node:20716) [MONGODB DRIVER] Warning: use
will be removed in the next major version
Server is running on port 3000
```

A screenshot of a terminal window titled 'C:\Program Files\MongoDB\'. It shows a MongoDB shell session where the user has switched to the 'usermanagement' database.

Test the Endpoints

Endpoints:

- User Endpoints:
 - POST /register - Register a new user

POST http://localhost:3000/us + ...

HTTP <http://localhost:3000/user/register> Save

POST <http://localhost:3000/user/register> Send

Params Authorization Headers (9) **Body** ● Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary **JSON** ▾ Beautify

```
1 {  
2   "username": "John",  
3   "password": "Password123",  
4   "role": "user"  
5 }  
6
```

Response



Click Send to get a response

Body Cookies Headers (7) Test Results

Status: 201 Created Time: 135 ms Size: 389 B Save Response ▾

Pretty Raw Preview Visualize

JSON ▾



```
1 {  
2   "username": "John",  
3   "password": "$2a$10$P8db2CbRX/RpVhL3CIsyR.2zsveuIhHw0Wsamh5nXI.6SBcT59n9C",  
4   "role": "user",  
5   "_id": "6706a096e68242d61d9f8cce",  
6   "__v": 0  
7 }
```

```
> db.users.find().pretty()  
{  
  "_id" : ObjectId("6706a096e68242d61d9f8cce"),  
  "username" : "John",  
  "password" : "$2a$10$P8db2CbRX/RpVhL3CIsyR.2zsveuIhHw0Wsamh5nXI.6SBcT59n9C",  
  "role" : "user",  
  "__v" : 0  
}  
> |
```

o POST /login - User login

POST http://localhost:3000/us + ⚡

HTTP <http://localhost:3000/user/login> [Save](#)

POST <http://localhost:3000/user/login> [Send](#) ↴

Params Authorization Headers (9) **Body** ● Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary JSON [Beautify](#)

```
1 {  
2   "username": "John",  
3   "password": "Password123",  
4   "role": "user"  
5 }  
6
```

Body Cookies Headers (7) Test Results [Save Response](#) ↴

Pretty Raw Preview Visualize [Q](#)

```
{"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NzA2YTA5NmU2ODI0MmQ2MWQ5ZjhjY2UiLCJyb2xIjoidXNlcjIwMjhdCI6MTcyODQ40DI2NH0.G8s1FNTh1RzjEV-yZPTRlxBYtS7ZysiCcA1KGID6k-Q"}
```

- o POST /upload - Upload an assignment

POST http://localhost:3000/user/upload

HTTP http://localhost:3000/user/upload Save

POST http://localhost:3000/user/upload Send

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary JSON Beautify

```
1 {  
2   "userId": "6706a096e68242d61d9f8cce",  
3   "task": "Hello World",  
4   "admin": "6706a50ae68242d61d9f8cd1"  
5 }  
6
```

Body Cookies Headers (7) Test Results Status: 201 Created Time: 12 ms Size: 433 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {  
2   "userId": "6706a096e68242d61d9f8cce",  
3   "task": "Hello World",  
4   "admin": "6706a50ae68242d61d9f8cd1",  
5   "status": "pending",  
6   "_id": "6706a6d0e68242d61d9f8cd3",  
7   "timestamp": "2024-10-09T15:52:48.101Z",  
8   "__v": 0  
9 }
```

```
> db.assignments.find().pretty()
{
    "_id" : ObjectId("6706a6d0e68242d61d9f8cd3"),
    "userId" : ObjectId("6706a096e68242d61d9f8cce"),
    "task" : "Hello World",
    "admin" : ObjectId("6706a50ae68242d61d9f8cd1"),
    "status" : "pending",
    "timestamp" : ISODate("2024-10-09T15:52:48.101Z"),
    "__v" : 0
}
```

- o GET /admins - fetch all admins

The screenshot shows the Postman application interface. At the top, there is a search bar with the text 'GET http://localhost:3000/use' and a '+' button for creating new requests. Below the search bar, the URL 'http://localhost:3000/user/admins' is entered into the 'HTTP' field. To the right, there is a 'Save' icon.

In the main request area, the method is set to 'GET' and the URL is 'http://localhost:3000/user/admins'. To the right of the URL, there is a large blue 'Send' button. Below the URL, there are tabs for 'Params', 'Authorization', 'Headers (7)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Body' tab is currently selected, indicated by a red underline. Under the 'Body' tab, there are five radio buttons: 'none' (selected), 'form-data', 'x-www-form-urlencoded', 'raw', and 'binary'. A note below the body tabs states 'This request does not have a body'.

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 11 ms Size: 386 B

Save Response ▾

Pretty Raw Preview Visualize

JSON ▾

```
1 []
2 {
3   "_id": "6706a50ae68242d61d9f8cd1",
4   "username": "Sam",
5   "password": "$2a$10$3UxXW0KVgQPgDlgyDS1w/.W7CzoqlMguttyMOxsE8aRGnAMz5tjXy",
6   "role": "admin",
7   "__v": 0
8 }
9 ]
```



Admin Endpoints:

- POST /register - Register a new admin

POST http://localhost:3000/ad



...

HTTP http://localhost:3000/admin/register

Save

POST



http://localhost:3000/admin/register

Send



Params

Authorization

Headers (9)

Body ●

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

JSON ▾

Beautify

```
1 {
2   ...
3   "username": "Sam",
4   "password": "Password1234",
5   "role": "admin"
6 }
```

Pretty Raw Preview Visualize

JSON ▾



```
1 {  
2   "username": "Sam",  
3   "password": "$2a$10$3UxXW0KVgQPgDlgyDS1w/.W7CzoqlMguttyMOXsE8aRGnAMz5tjXy",  
4   "role": "admin",  
5   "_id": "6706a50ae68242d61d9f8cd1",  
6   "__v": 0  
7 }
```

```
> db.users.find().pretty()
```

```
{  
  "_id" : ObjectId("6706a50ae68242d61d9f8cd1"),  
  "username" : "Sam",  
  "password" : "$2a$10$3UxXW0KVgQPgDlgyDS1w/.W7CzoqlMguttyMOXsE8aRGnAMz5tjXy",  
  "role" : "admin",  
  "__v" : 0  
}
```

- o POST /login - Admin login

POST http://localhost:3000/ad + ⚙

HTTP <http://localhost:3000/admin/login> Save

POST http://localhost:3000/admin/login Send

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary JSON Beautify

```
1 {  
2   "username": "Sam",  
3   "password": "Password1234",  
4   "role": "admin"  
5 }  
6
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 81 ms Size: 417 B Save Response

Pretty Raw Preview Visualize

```
{"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOjI2NzA2YTUwYWU2ODI0MmQ2MWQ5ZjhjZDEiLCJyb2xIjoIYWRtaW4iLCJpYXQiOjE3Mjg0ODk3Mzd9.W2WQY-dZ1z1lN8vFDr40THniEV7K-xfX8-vNk7Bgixk"}
```

- GET /assignments - View assignments tagged to the admin
 - > show collections
 - assignments
 - users

GET http://localhost:3000/admin

HTTP http://localhost:3000/admin/assignments?adminId=6706a50ae68242d61d9f8cd1 Save

GET http://localhost:3000/admin/assignments?adminId=6706a50ae68242d61d9f8cd1 Send

Params • Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary

This request does not have a body

Body Cookies Headers (7) Test Results Status: 200 OK Time: 9 ms Size: 430 B Save Response

Pretty Raw Preview Visualize JSON

```
1
2   {
3     "_id": "6706a6d0e68242d61d9f8cd3",
4     "userId": "6706a096e68242d61d9f8cce",
5     "task": "Hello World",
6     "admin": "6706a50ae68242d61d9f8cd1",
7     "status": "pending",
8     "timestamp": "2024-10-09T15:52:48.101Z",
9     "__v": 0
10  }
11
```

- POST /assignments/:id/accept - Accept an assignment

POST http://localhost:3000/ad + ⚙

HTTP <http://localhost:3000/admin/assignments/6706a6d0e68242d61d9f8cd3/accept> Save

POST <http://localhost:3000/admin/assignments/6706a6d0e68242d61d9f8cd3/accept> Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Body Cookies Headers (7) Test Results Status: 200 OK Time: 13 ms Size: 429 B Save Response

Pretty Raw Preview Visualize JSON ↻

```
1 {  
2   "_id": "6706a6d0e68242d61d9f8cd3",  
3   "userId": "6706a096e68242d61d9f8cce",  
4   "task": "Hello World",  
5   "admin": "6706a50ae68242d61d9f8cd1",  
6   "status": "accepted",  
7   "timestamp": "2024-10-09T15:52:48.101Z",  
8   "__v": 0  
9 }
```

```
> db.assignments.find().pretty()
{
    "_id" : ObjectId("6706a6d0e68242d61d9f8cd3"),
    "userId" : ObjectId("6706a096e68242d61d9f8cce"),
    "task" : "Hello World",
    "admin" : ObjectId("6706a50ae68242d61d9f8cd1"),
    "status" : "accepted",
    "timestamp" : ISODate("2024-10-09T15:52:48.101Z"),
    "__v" : 0
}
```

- o POST /assignments/:id/reject - Reject an assignment

The screenshot shows the Postman application interface. At the top, there is a header bar with a 'POST' button, a URL input field containing 'http://localhost:3000/admin/assignments/6706a6d0e68242d61d9f8cd3/reject', and a 'Save' button. Below the header, the main request configuration area is visible, featuring a 'POST' method dropdown and a URL input field with the same endpoint. To the right of this area is a large blue 'Send' button. Underneath the request configuration, there are several tabs: 'Params' (which is selected), 'Authorization', 'Headers (8)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Params' tab contains a 'Query Params' section with a table. This table has two rows: the first row has a 'Key' column with 'Key' and a 'Value' column with 'Value'; the second row also has a 'Key' column with 'Key' and a 'Value' column with 'Value'. There is also a 'Bulk Edit' button at the bottom right of the table.

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 12 ms Size: 429 B | Save Response ▾

Pretty

Raw

Preview

Visualize

JSON ▾



```
1 {  
2   "_id": "6706a6d0e68242d61d9f8cd3",  
3   "userId": "6706a096e68242d61d9f8cce",  
4   "task": "Hello World",  
5   "admin": "6706a50ae68242d61d9f8cd1",  
6   "status": "rejected",  
7   "timestamp": "2024-10-09T15:52:48.101Z",  
8   "__v": 0  
9 }
```

```
> db.assignments.find().pretty()  
{  
    "_id" : ObjectId("6706a6d0e68242d61d9f8cd3"),  
    "userId" : ObjectId("6706a096e68242d61d9f8cce"),  
    "task" : "Hello World",  
    "admin" : ObjectId("6706a50ae68242d61d9f8cd1"),  
    "status" : "rejected",  
    "timestamp" : ISODate("2024-10-09T15:52:48.101Z")  
    "__v" : 0  
}
```