

Name: *Chinnakotla Jagannath*

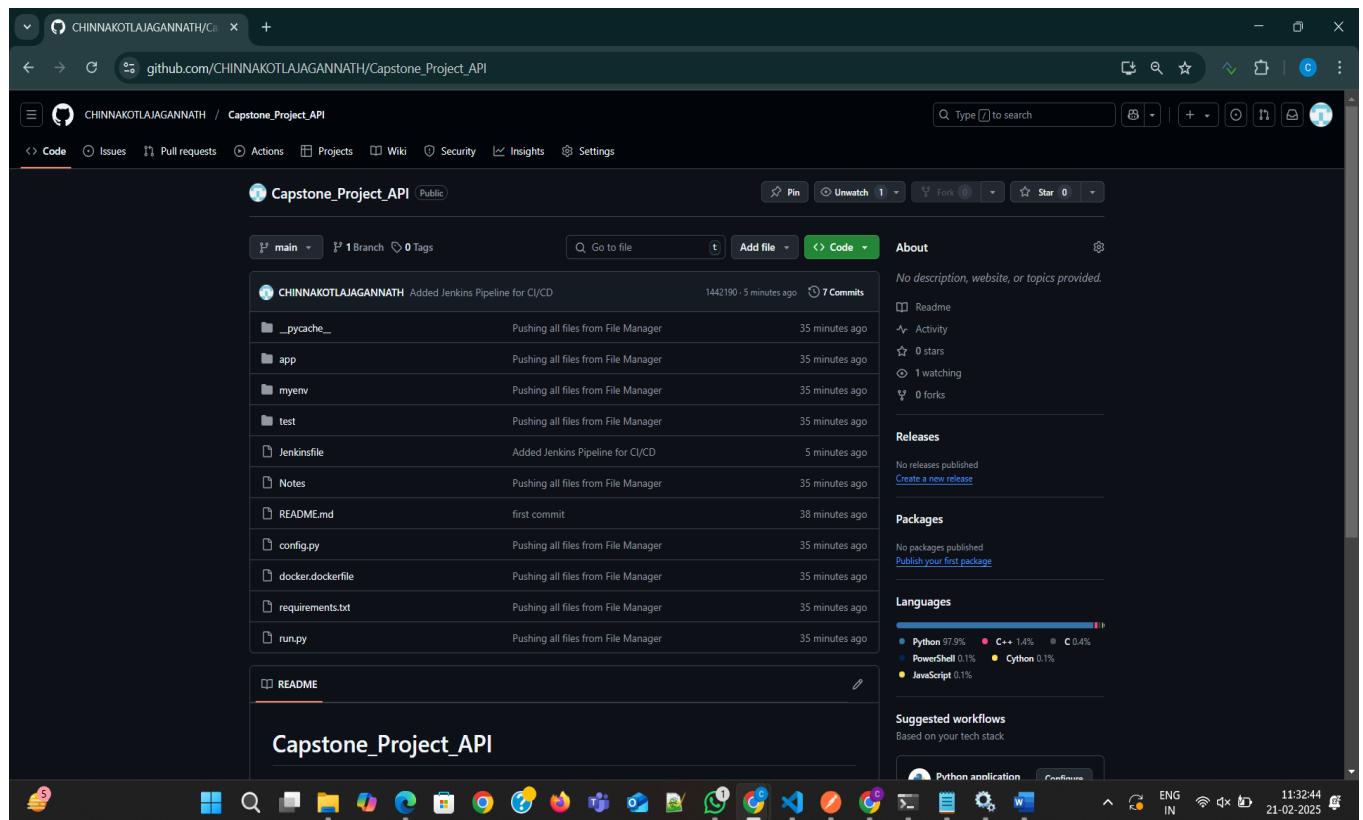
## Capstone Project: *DevOps-Driven Flask Application with SQLite and Azure Deployment*

### Objectives

#### 1. Develop a Flask API

- a. Design a RESTful API that allows users to perform CRUD operations.
- b. Implement authentication and error handling mechanisms.

**Github\_Link:** [https://github.com/CHINNAKOTLAJAGANNATH/Capstone\\_Project\\_API.git](https://github.com/CHINNAKOTLAJAGANNATH/Capstone_Project_API.git)



**Run:** Running on <http://127.0.0.1:5000>

#### 1. User Management API

➤ [Login](#)

The screenshot shows the Postman application interface. A GET request is made to `http://127.0.0.1:5000`. The response status is `200 OK` with a response time of `14 ms` and a size of `196 B`. The response body contains the message `Welcome to the User API`.

➤ **POST /users → Add a new user.**  
<http://127.0.0.1:5000/users>

The screenshot shows the Postman application interface. A POST request is made to `http://127.0.0.1:5000/users` with the following JSON body:

```
1 {
2   "username": "haisha",
3   "password": 12345
4 }
```

The response status is `201 CREATED` with a response time of `36 ms` and a size of `213 B`. The response body is:

```
1 {
2   "id": 6,
3   "message": "User created"
4 }
```

➤ **GET /users → Retrieve a list of all users.**  
<http://127.0.0.1:5000/users>

The screenshot shows the Postman interface with a successful API call. The URL is `http://127.0.0.1:5000/users`. The response body is a JSON array containing five user objects:

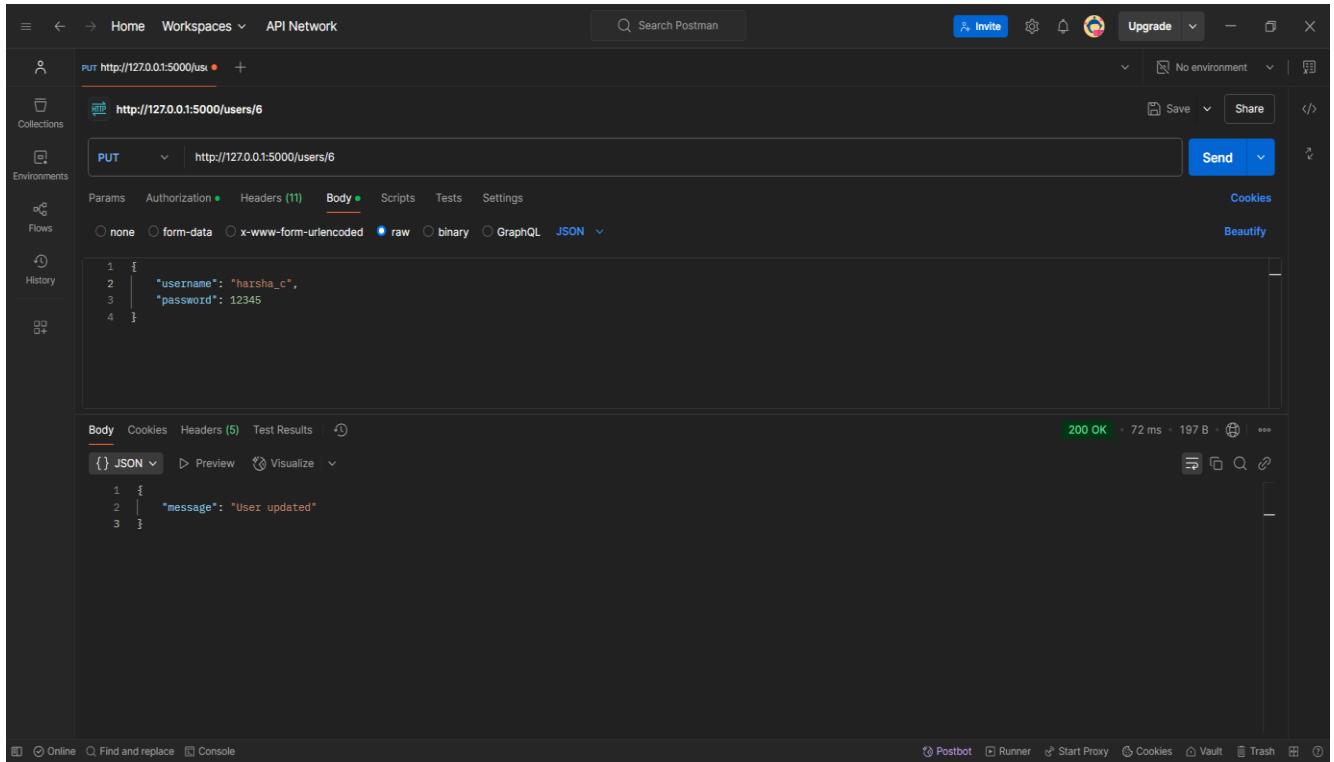
```
[{"id": 1, "username": "ram"}, {"id": 2, "username": "jagan"}, {"id": 3, "username": "james"}, {"id": 4, "username": "john doe"}]
```

- **Get / users/{id} → Retrieve details of a specific user.**  
<http://127.0.0.1:5000/users/3>

The screenshot shows the Postman interface with a successful API call. The URL is `http://127.0.0.1:5000/users/3`. The response body is a JSON object representing a single user:

```
{"id": 3, "username": "jagan"}
```

- **PUT /users/{id} → Update user details.**  
<http://127.0.0.1:5000/users/6>



The screenshot shows the Postman interface with a dark theme. A collection named 'Users' is selected on the left sidebar. In the main workspace, a PUT request is made to <http://127.0.0.1:5000/users/6>. The 'Body' tab is selected, showing raw JSON data:

```

1 {
2   "username": "harsha_c",
3   "password": "12345"
4 }

```

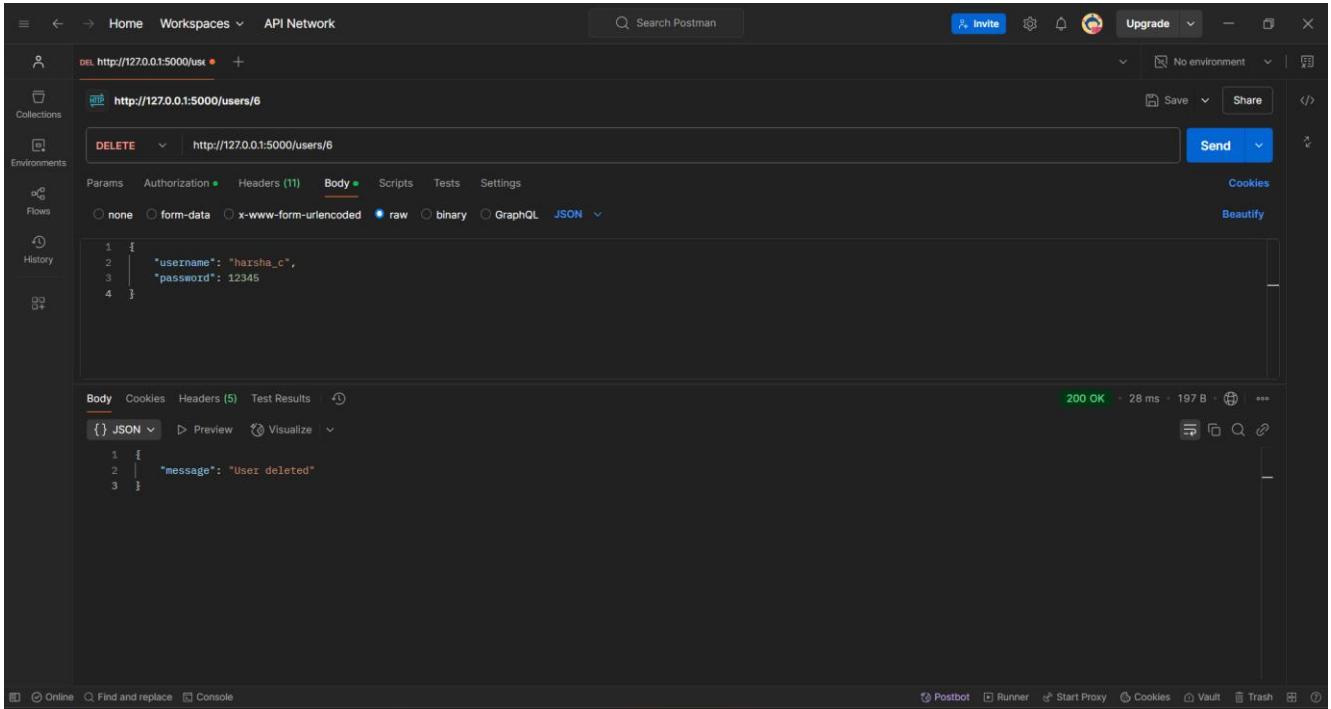
The response status is 200 OK, with a response time of 72 ms and a body size of 197 B. The response body is:

```

1 {}
2   "message": "User updated"
3

```

- **DELETE /users/{id} → Delete a user from the database.**  
<http://127.0.0.1:5000/users/6>



The screenshot shows the Postman interface with a dark theme. A collection named 'Users' is selected on the left sidebar. In the main workspace, a DELETE request is made to <http://127.0.0.1:5000/users/6>. The 'Body' tab is selected, showing raw JSON data:

```

1 {
2   "username": "harsha_c",
3   "password": "12345"
4 }

```

The response status is 200 OK, with a response time of 28 ms and a body size of 197 B. The response body is:

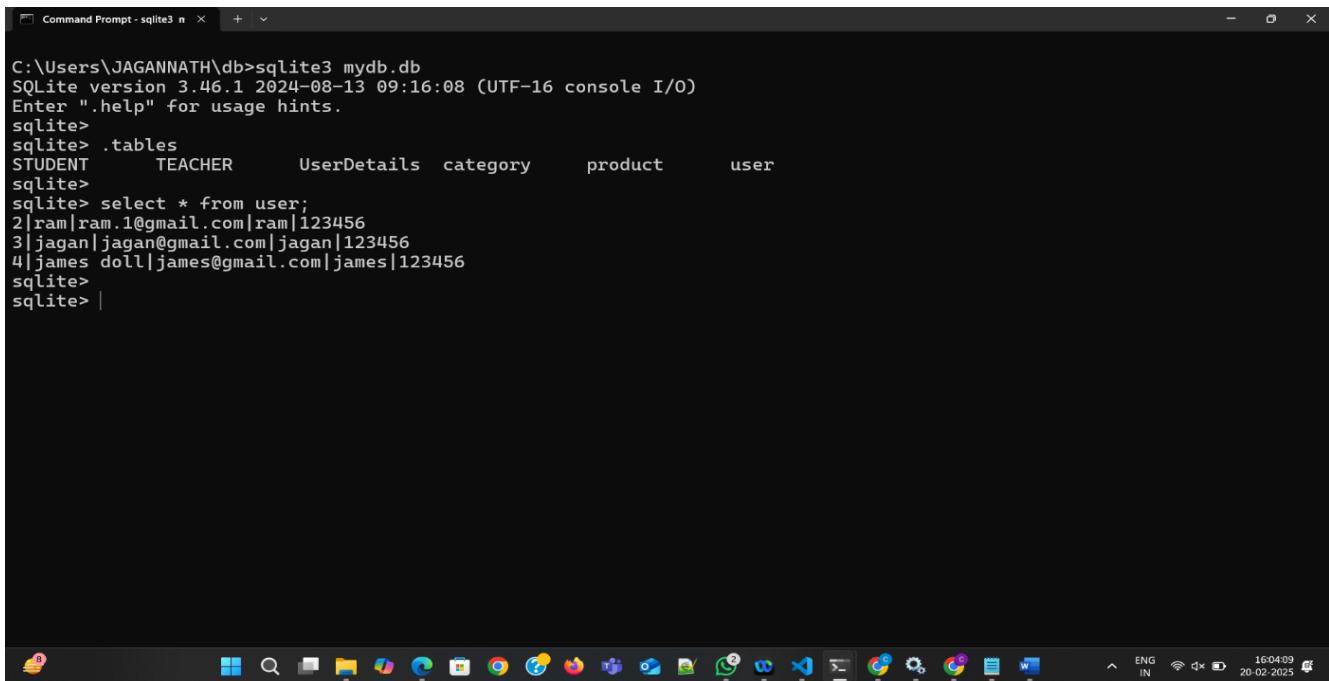
```

1 {}
2   "message": "User deleted"
3

```

## 2. Integrate an SQLite Database

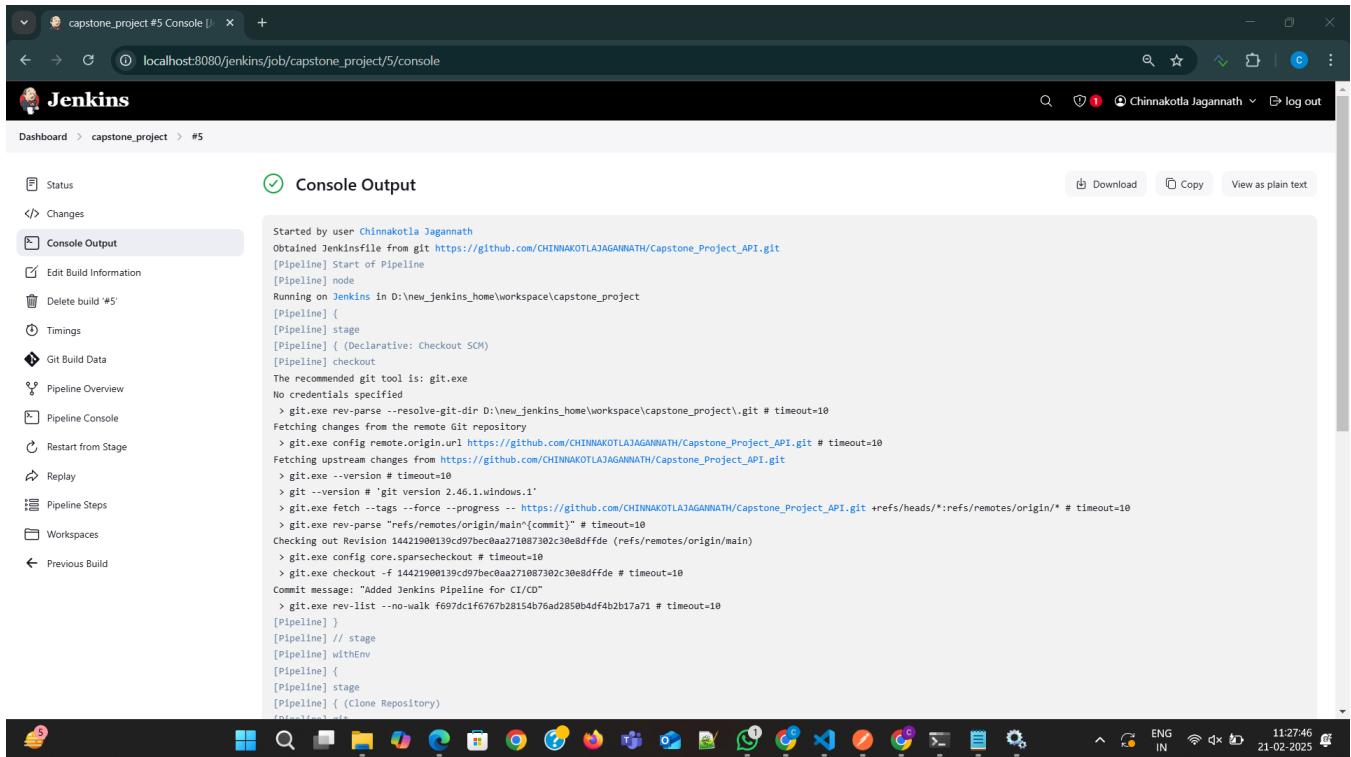
- Create and manage a database using SQLite.
- Store and retrieve structured data efficiently.



```
C:\Users\JAGANNATH\db>sqlite3 mydb.db
SQLite version 3.46.1 2024-08-13 09:16:08 (UTF-16 console I/O)
Enter ".help" for usage hints.
sqlite>.tables
STUDENT      TEACHER      UserDetails  category    product      user
sqlite>
sqlite>.select * from user;
2|ram|ram.1@gmail.com|ram|123456
3|jagan|jagan@gmail.com|jagan|123456
4|james doll|james@gmail.com|james|123456
sqlite>|
sqlite> |
```

### 3. Automate DevOps Pipeline Using Jenkins

- Set up continuous integration and deployment (CI/CD) pipelines.
- Automate building, testing, and deploying the Flask application.



The screenshot shows the Jenkins interface for a job named "capstone\_project #5". The "Console Output" tab is selected, displaying the build logs. The logs show the Jenkinsfile being obtained via Git, the pipeline starting, and the execution of various Jenkins steps like "git", "config", "fetch", and "rev-parse". The Jenkins UI includes a sidebar with links like Status, Changes, and Pipeline Overview, and a header with navigation and user information.

```
Started by user Chinnakotla Jagannath
Obtained Jenkinsfile from git https://github.com/CHINNAKOTLAJAGANNATH/Capstone_Project_API.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in D:\new_jenkins_home\workspace\capstone_project
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
The recommended git tool is: git.exe
No credentials specified
> git.exe rev-parse --resolve-git-dir D:\new_jenkins_home\workspace\capstone_project\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/CHINNAKOTLAJAGANNATH/Capstone_Project_API.git # timeout=10
Fetching upstream changes from https://github.com/CHINNAKOTLAJAGANNATH/Capstone_Project_API.git
> git.exe -version # timeout=10
> git.exe checkout -f 14421900139cd97bec0aa27108730c2c30e8dffde # timeout=10
Commit message: "Added Jenkins Pipeline for CI/CD"
> git.exe rev-list --no-walk f697dc1f6767b28154b7ead2850b4df4b2b17a71 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Clone Repository)
[Pipeline] }
```

The screenshot shows a Jenkins pipeline console output for a project named 'capstone\_project'. The output details the steps taken during the build process, including cloning the repository, fetching upstream changes, and committing the changes. The final message indicates a successful build.

```
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Clone Repository)
[Pipeline] git
The recommended git tool is: git.exe
using credential cred-jenkins
> git.exe rev-parse --resolve-git-dir D:\new_jenkins_home\workspace\capstone_project\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/CHINNAKOTLAJAGANNATH/Capstone_Project_API.git # timeout=10
Fetching upstream changes from https://github.com/CHINNAKOTLAJAGANNATH/Capstone_Project_API.git
> git.exe -version # timeout=10
> git --version # 'git version 2.46.1.windows.1'
using GIT_ASKPASS to set credentials
> git.exe fetch --tags --force --progress -- https://github.com/CHINNAKOTLAJAGANNATH/Capstone_Project_API.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision 14421900139cd97bec0aa271087302c30e8dffde (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 14421900139cd97bec0aa271087302c30e8dffde # timeout=10
> git.exe branch -a -v --no-abbrev # timeout=10
> git.exe branch -D main # timeout=10
> git.exe checkout -b main 14421900139cd97bec0aa271087302c30e8dffde # timeout=10
Commit message: "Added Jenkins Pipeline for CI/CD"
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

