NAME: SRI HARI DHEERAJ KOMMINENI
CWID: 885177196

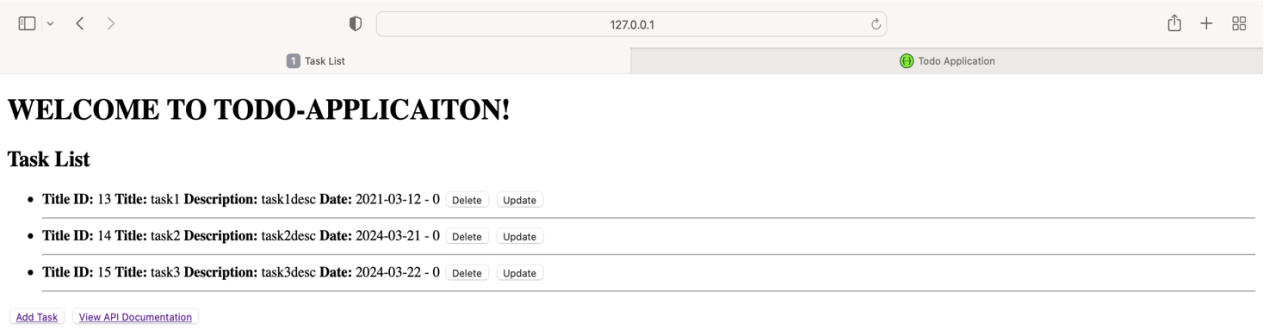
CPSC 449 WEB-BACKEND ASSIGNMENT

TO-DO FLASK APPLICATION SCREENSHOTS

This ToDo Application has two pages, Home page (index.html) which shows all the tasks and allows us to either delete a task or update it and Add Task Page(add_task.html) which allows us to add a task.

The following are the screenshots of the application pages.

- **Application Home page (index.html)**



- **Add Task Page (add_task.html)**

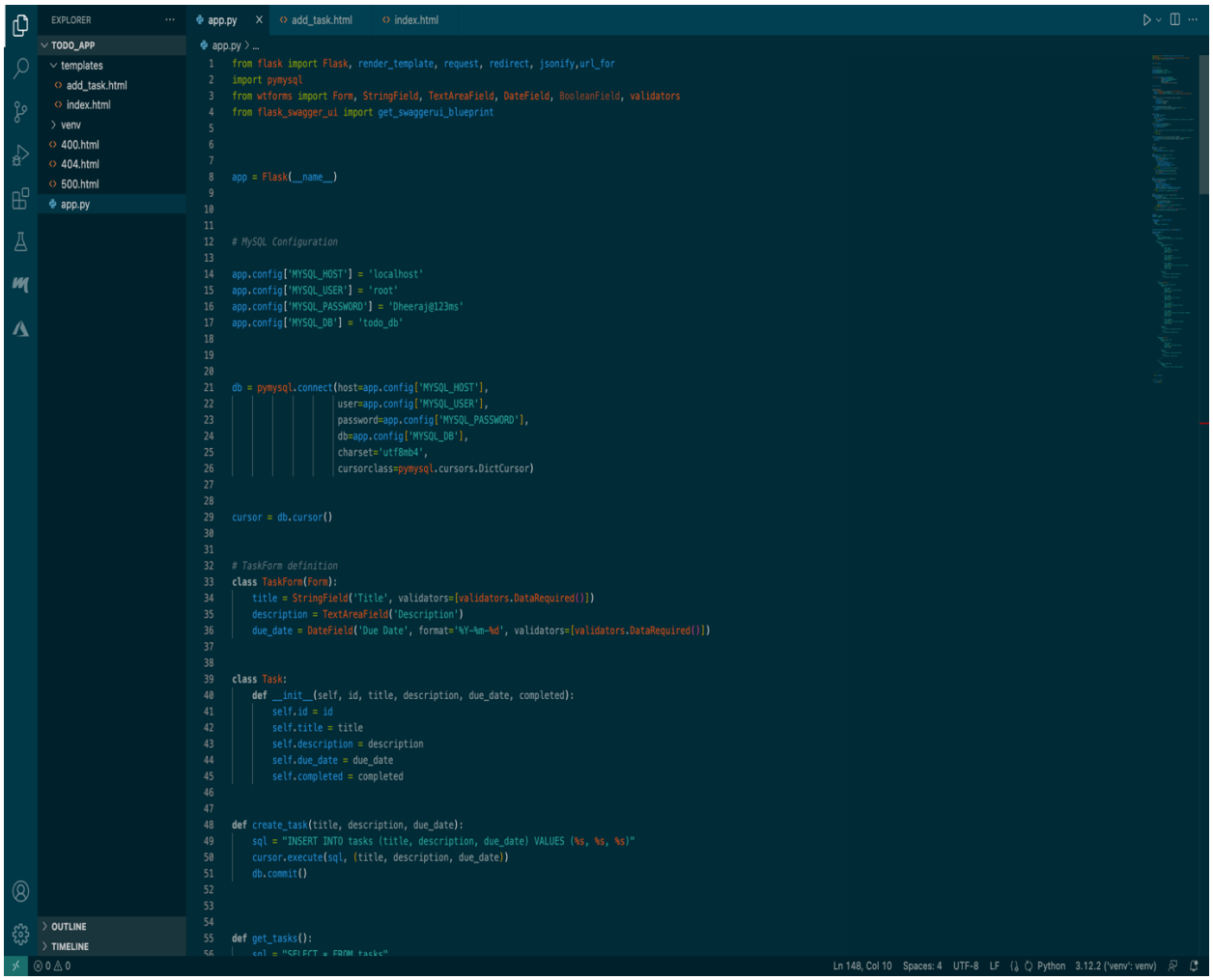
The following is the Database table schema and entries that have been used in the application (todo_db) and table (tasks)

```
+------------------+
| Tables_in_todo_db |
+------------------+
| tasks            |
+------------------+
1 row in set (0.01 sec)

mysql> SELCT * FROM tasks;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'SELCT * FROM tasks' at line 1
mysql> DESCRIBE todo_db.tasks;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| id          | int          | NO   | PRI | NULL    | auto_increment |
| title       | varchar(255) | NO   |     | NULL    |                |
| description | text         | YES  |     | NULL    |                |
| due_date    | date         | YES  |     | NULL    |                |
| completed   | tinyint(1)   | YES  |     | 0       |                |
+-------------+--------------+------+-----+---------+----------------+
5 rows in set (0.01 sec)

mysql> SHOW COLUMNS FROM todo_db.tasks;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| id          | int          | NO   | PRI | NULL    | auto_increment |
| title       | varchar(255) | NO   |     | NULL    |                |
| description | text         | YES  |     | NULL    |                |
| due_date    | date         | YES  |     | NULL    |                |
| completed   | tinyint(1)   | YES  |     | 0       |                |
+-------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM todo_db.tasks;
+----+--------------+-----------------------+------------+-----------+
| id | title        | description           | due_date   | completed |
+----+--------------+-----------------------+------------+-----------+
|  2 | Task updated | updated description | 2024-03-17 |         1 |
+----+--------------+-----------------------+------------+-----------+
1 row in set (0.00 sec)

mysql> SELECT * FROM todo_db.tasks;
+----+----------------+-----------------------+------------+-----------+
| id | title          | description           | due_date   | completed |
+----+----------------+-----------------------+------------+-----------+
|  2 | Task updated   | updated description | 2024-03-17 |         1 |
|  7 | task updated 2 | task updated 2 desc | 2024-03-12 |         0 |
+----+----------------+-----------------------+------------+-----------+
2 rows in set (0.00 sec)

mysql> USE todo_db
Database changed
mysql> SELECT * FROM todo_db.tasks;
+----+---------+--------------+------------+-----------+
| id | title   | description | due_date   | completed |
+----+---------+--------------+------------+-----------+
| 13 | task1   | task1desc    | 2021-03-12 |         0 |
| 16 | task3   | task3desc    | 2024-03-17 |         0 |
| 17 | task 6 | task 6 desc | 2024-03-02 |         0 |
+----+---------+--------------+------------+-----------+
3 rows in set (0.00 sec)

mysql>
```

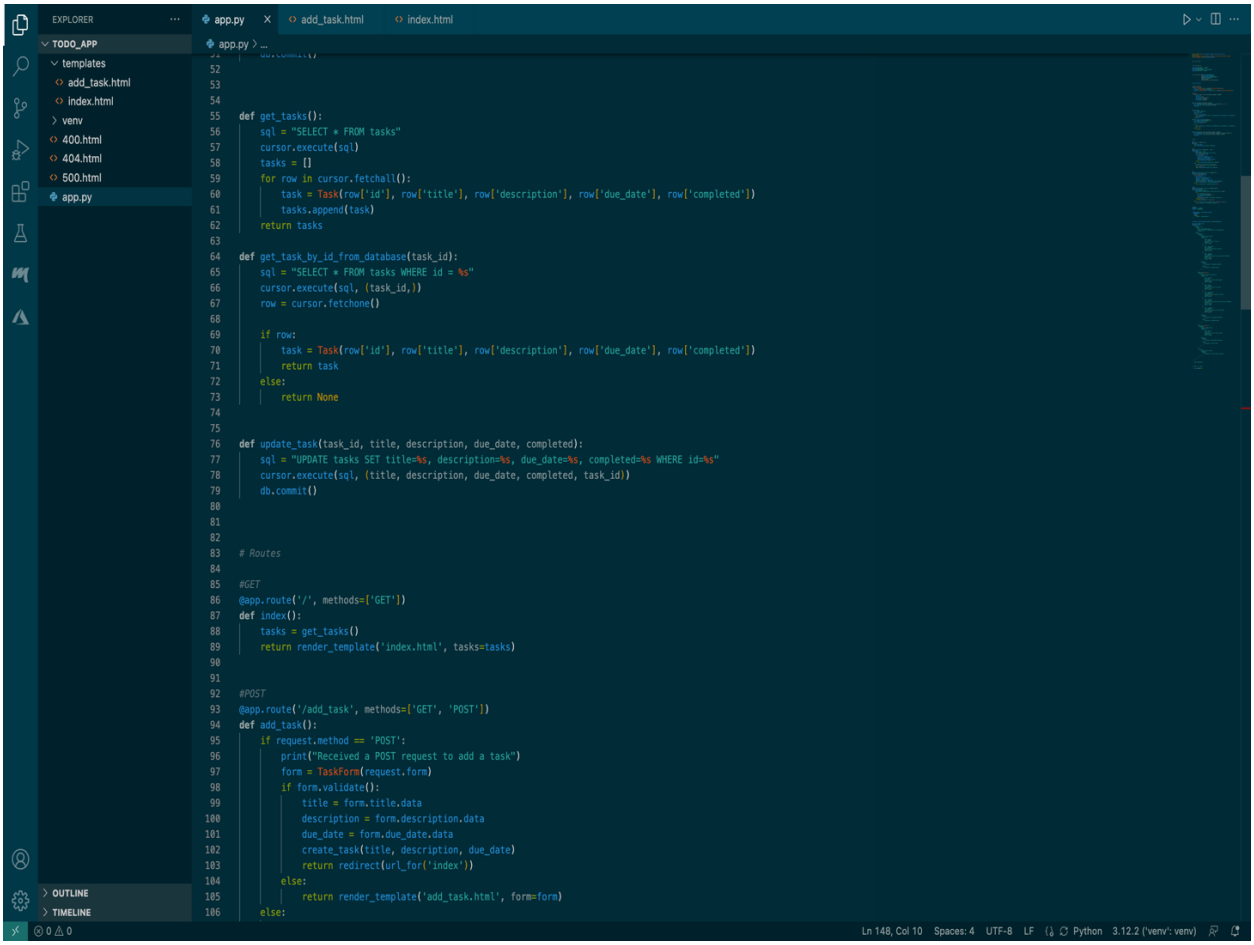
The following are my app.py where I did all the API endpoint definitions, index.html and add_task.html which are the Frontend pages of the applications.

- **app.py code**

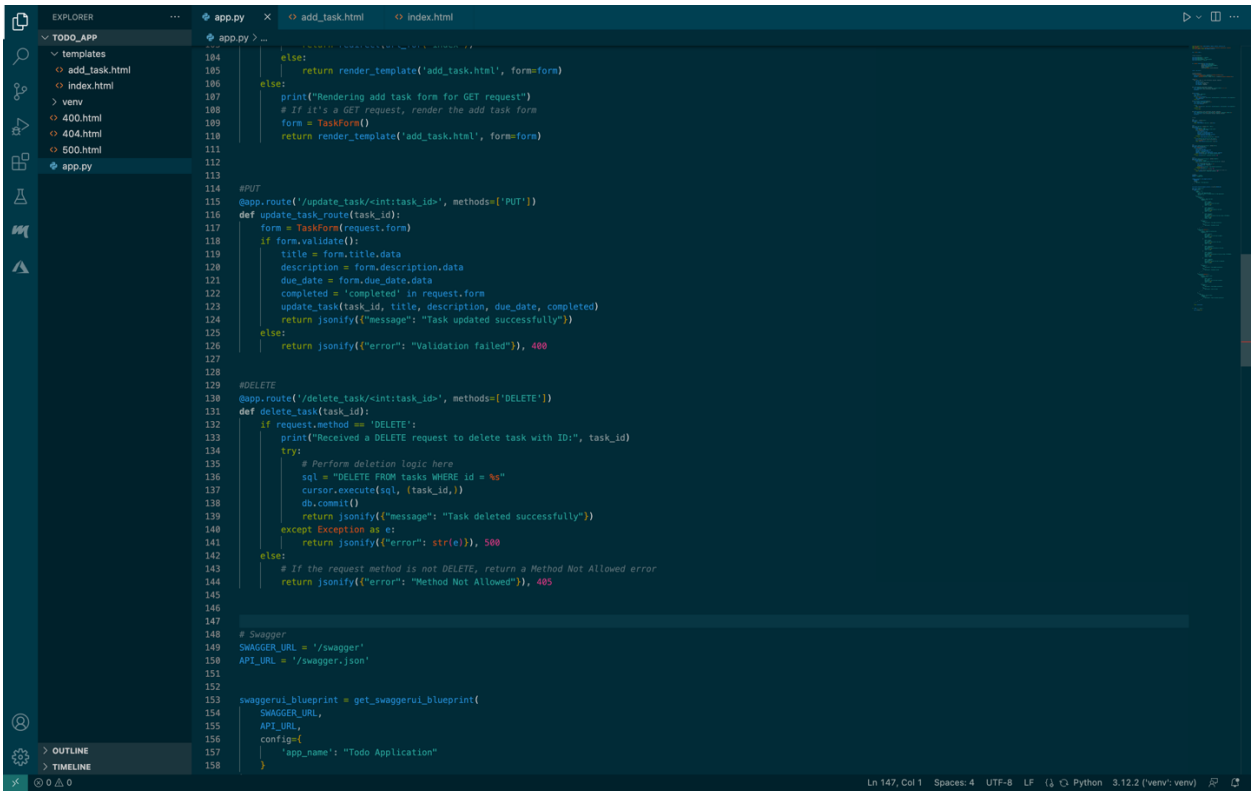
NAME: SRI HARI DHEERAJ KOMMINENI
CWID: 885177196

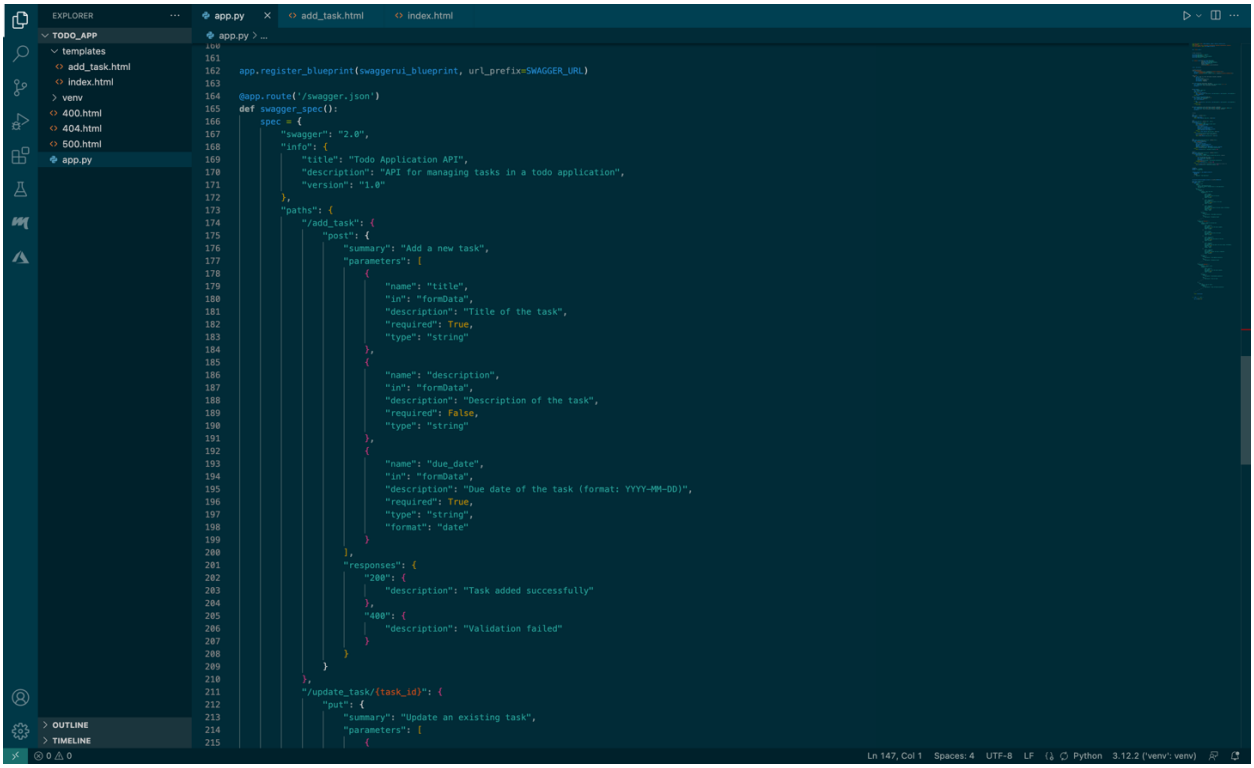


NAME: SRI HARI DHEERAJ KOMMINENI
CWID: 885177196

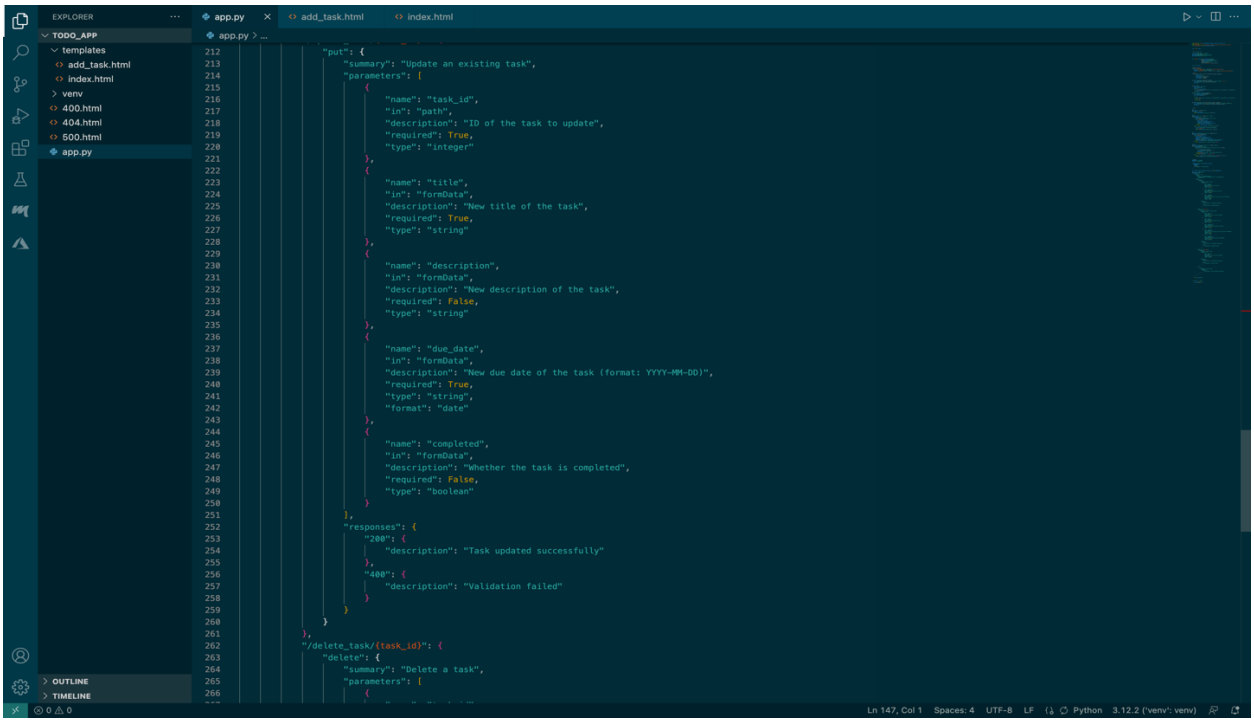


NAME: SRI HARI DHEERAJ KOMMINENI
CWID: 885177196

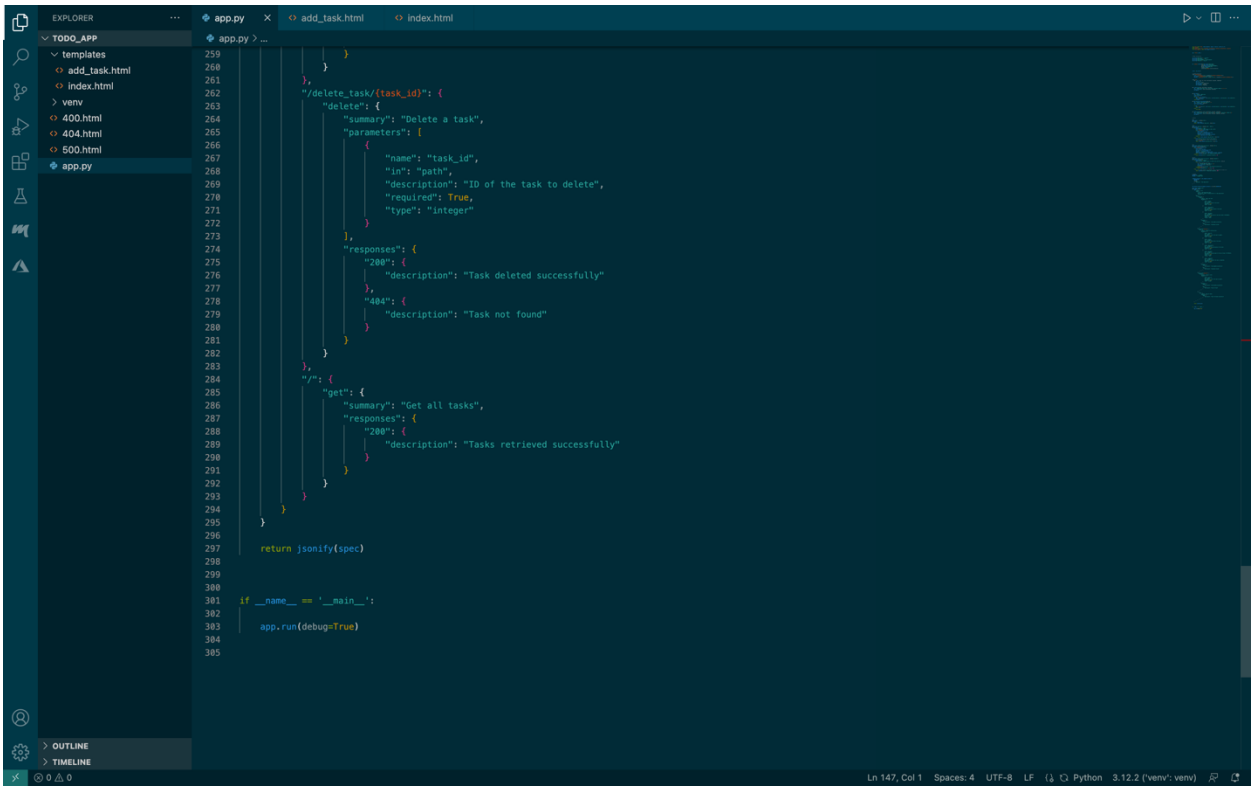




NAME: SRI HARI DHEERAJ KOMMINENI
CWID: 885177196



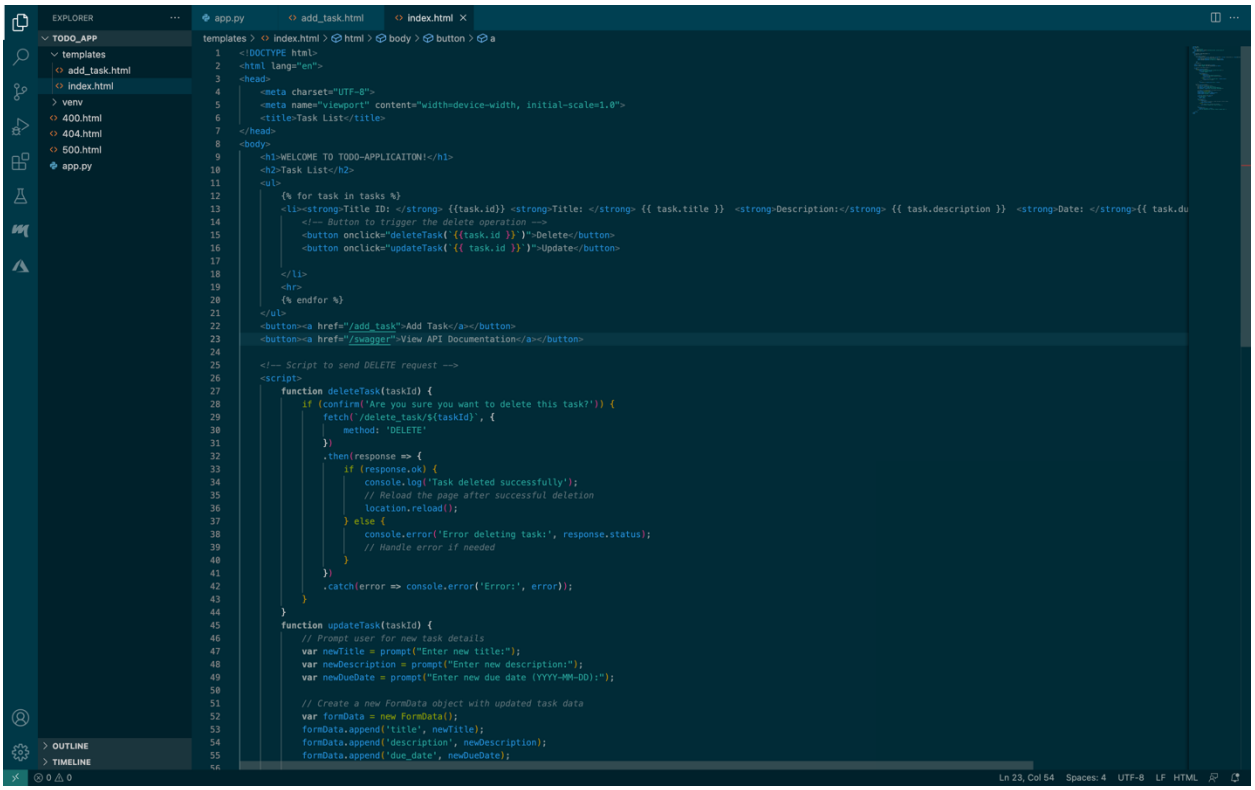
```
            "put": {
                "summary": "Update an existing task",
                "parameters": [
                    {
                        "name": "task_id",
                        "in": "path",
                        "description": "ID of the task to update",
                        "required": True,
                        "type": "integer"
                    },
                    {
                        "name": "title",
                        "in": "formData",
                        "description": "New title of the task",
                        "required": True,
                        "type": "string"
                    },
                    {
                        "name": "description",
                        "in": "formData",
                        "description": "New description of the task",
                        "required": False,
                        "type": "string"
                    },
                    {
                        "name": "due_date",
                        "in": "formData",
                        "description": "New due date of the task (format: YYYY-MM-DD)",
                        "required": True,
                        "type": "string",
                        "format": "date"
                    },
                    {
                        "name": "completed",
                        "in": "formData",
                        "description": "Whether the task is completed",
                        "required": False,
                        "type": "boolean"
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Task updated successfully"
                    },
                    "400": {
                        "description": "Validation failed"
                    }
                }
            }
        },
        "/delete_task/{task_id}": {
            "delete": {
                "summary": "Delete a task",
                "parameters": [
                    {
```



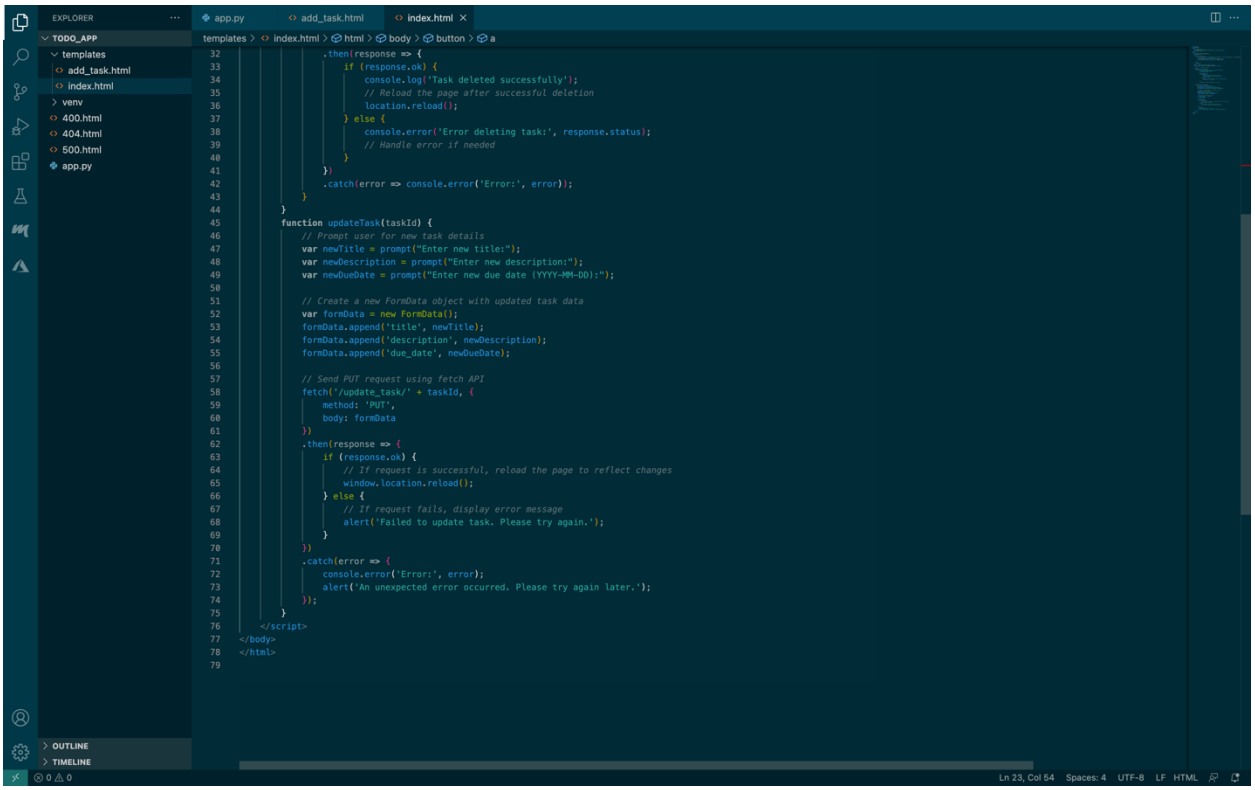
```
            }
        },
        "/delete_task/{task_id}": {
            "delete": {
                "summary": "Delete a task",
                "parameters": [
                    {
                        "name": "task_id",
                        "in": "path",
                        "description": "ID of the task to delete",
                        "required": True,
                        "type": "integer"
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Task deleted successfully"
                    },
                    "404": {
                        "description": "Task not found"
                    }
                }
            }
        },
        "/": {
            "get": {
                "summary": "Get all tasks",
                "responses": {
                    "200": {
                        "description": "Tasks retrieved successfully"
                    }
                }
            }
        }
    }
}

    return jsonify(spec)


if __name__ == '__main__':

    app.run(debug=True)
```

NAME: SRI HARI DHEERAJ KOMMINENI
CWID: 885177196

- **index.html (Home Page)**

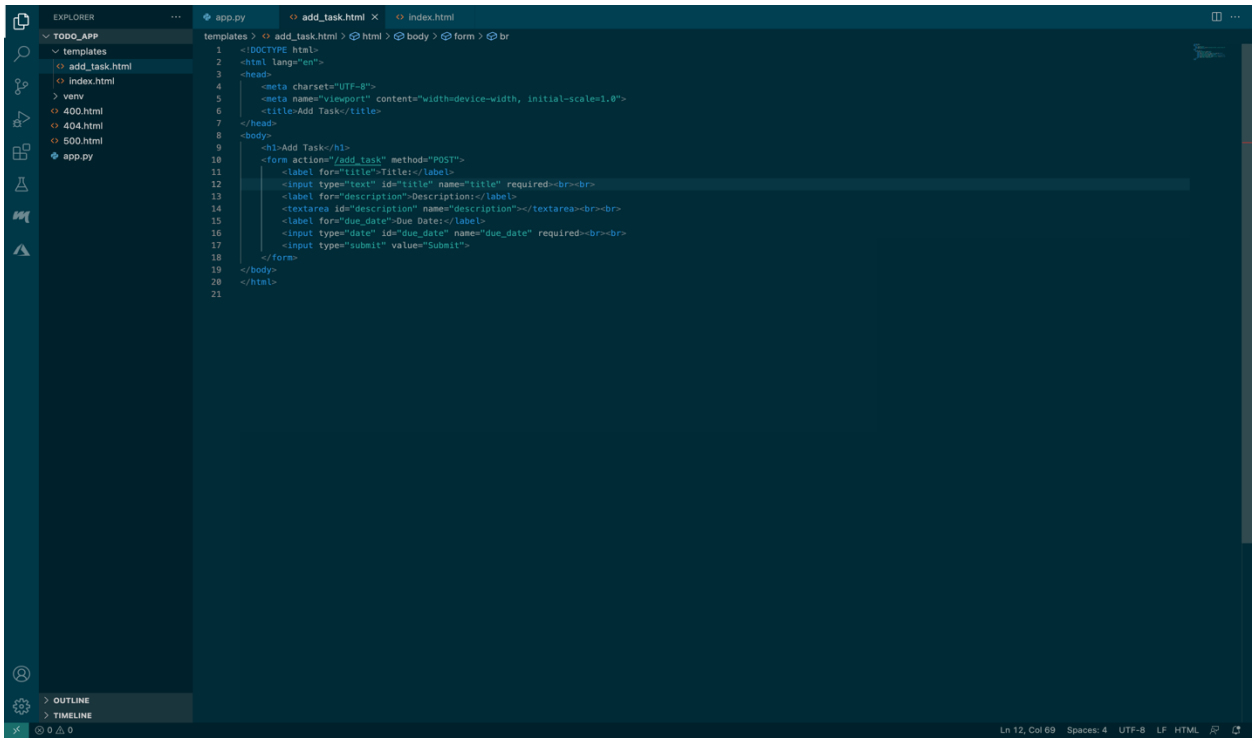


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task List</title>
</head>
<body>
    <h1>WELCOME TO TODO-APPLICAITON!</h1>
    <h2>Task List</h2>
    <ul>
        {% for task in tasks %}
        <li><strong>Title ID: </strong> {{task.id}} <strong>Title: </strong> {{ task.title }}  <strong>Description:</strong> {{ task.description }}  <strong>Date: </strong>{{ task.du
            <!-- Button to trigger the delete operation -->
            <button onclick="deleteTask(`{{task.id }}`)">Delete</button>
            <button onclick="updateTask(`{{ task.id }}`)">Update</button>

        </li>
        <hr>
        {% endfor %}
    </ul>
    <button><a href="/add_task">Add Task</a></button>
    <button><a href="/swagger">View API Documentation</a></button>

    <!-- Script to send DELETE request -->
    <script>
        function deleteTask(taskId) {
            if (confirm('Are you sure you want to delete this task?')) {
                fetch(`/delete_task/${taskId}`, {
                    method: 'DELETE'
                })
                .then(response => {
                    if (response.ok) {
                        console.log('Task deleted successfully');
                        // Reload the page after successful deletion
                        location.reload();
                    } else {
                        console.error('Error deleting task:', response.status);
                        // Handle error if needed
                    }
                })
                .catch(error => console.error('Error:', error));
            }
        }
        function updateTask(taskId) {
            // Prompt user for new task details
            var newTitle = prompt("Enter new title:");
            var newDescription = prompt("Enter new description:");
            var newDueDate = prompt("Enter new due date (YYYY-MM-DD):");

            // Create a new FormData object with updated task data
            var formData = new FormData();
            formData.append('title', newTitle);
            formData.append('description', newDescription);
            formData.append('due_date', newDueDate);
```



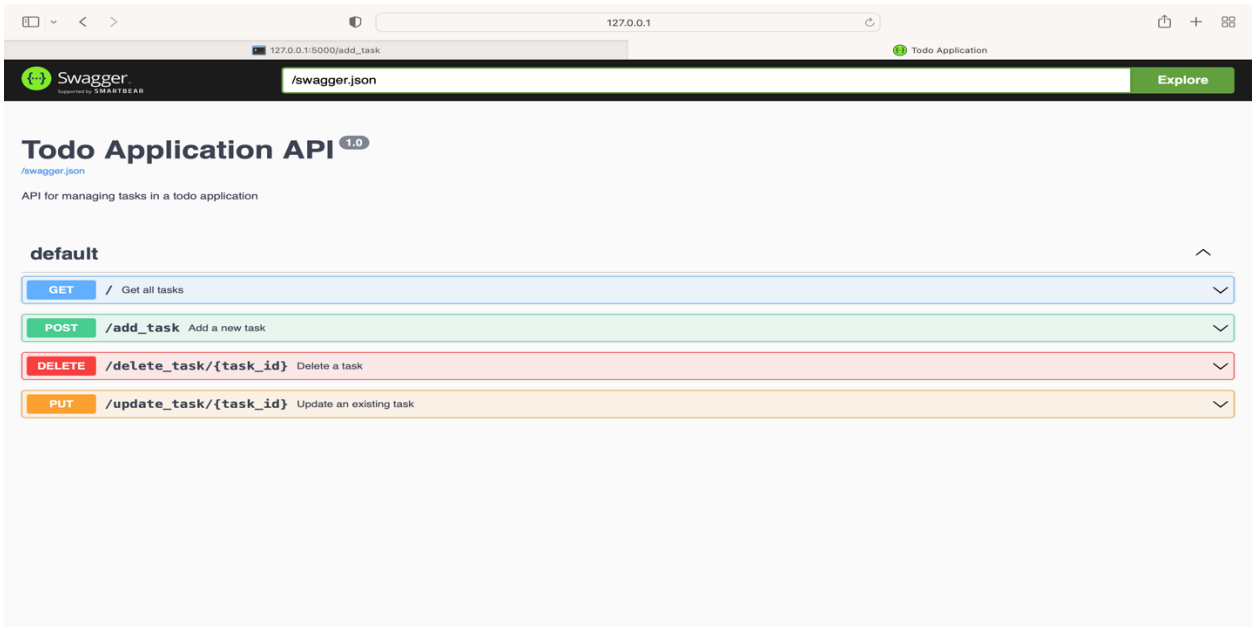
```
                .then(response => {
                    if (response.ok) {
                        console.log('Task deleted successfully');
                        // Reload the page after successful deletion
                        location.reload();
                    } else {
                        console.error('Error deleting task:', response.status);
                        // Handle error if needed
                    }
                })
                .catch(error => console.error('Error:', error));
            }
        }
        function updateTask(taskId) {
            // Prompt user for new task details
            var newTitle = prompt("Enter new title:");
            var newDescription = prompt("Enter new description:");
            var newDueDate = prompt("Enter new due date (YYYY-MM-DD):");

            // Create a new FormData object with updated task data
            var formData = new FormData();
            formData.append('title', newTitle);
            formData.append('description', newDescription);
            formData.append('due_date', newDueDate);

            // Send PUT request using fetch API
            fetch('/update_task/' + taskId, {
                method: 'PUT',
                body: formData
            })
            .then(response => {
                if (response.ok) {
                    // If request is successful, reload the page to reflect changes
                    window.location.reload();
                } else {
                    // If request fails, display error message
                    alert('Failed to update task. Please try again.');
                }
            })
            .catch(error => {
                console.error('Error:', error);
                alert('An unexpected error occurred. Please try again later.');
            });
        }
    </script>
</body>
</html>
```

- **add_task.html (Add Task Page)**



Following is the screenshot of Swagger UI of my application which has all the GET, POST, PUT, DELETE definitions.

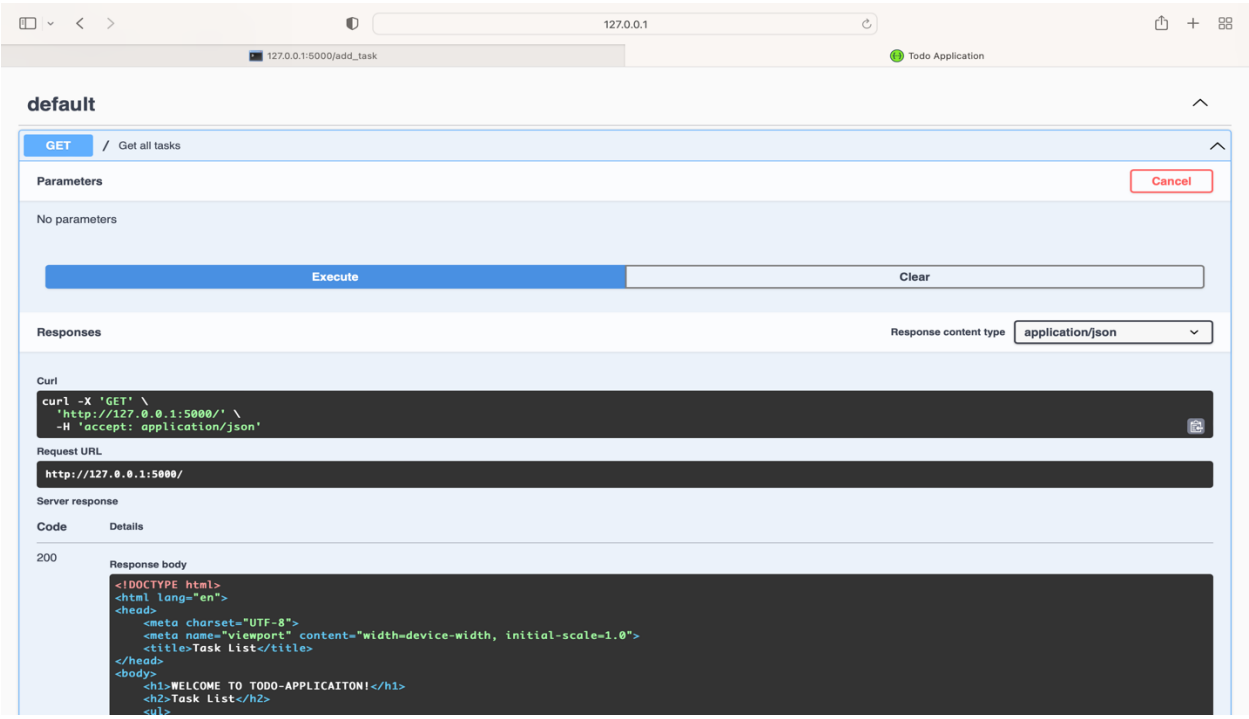
- **SWAGGER UI**

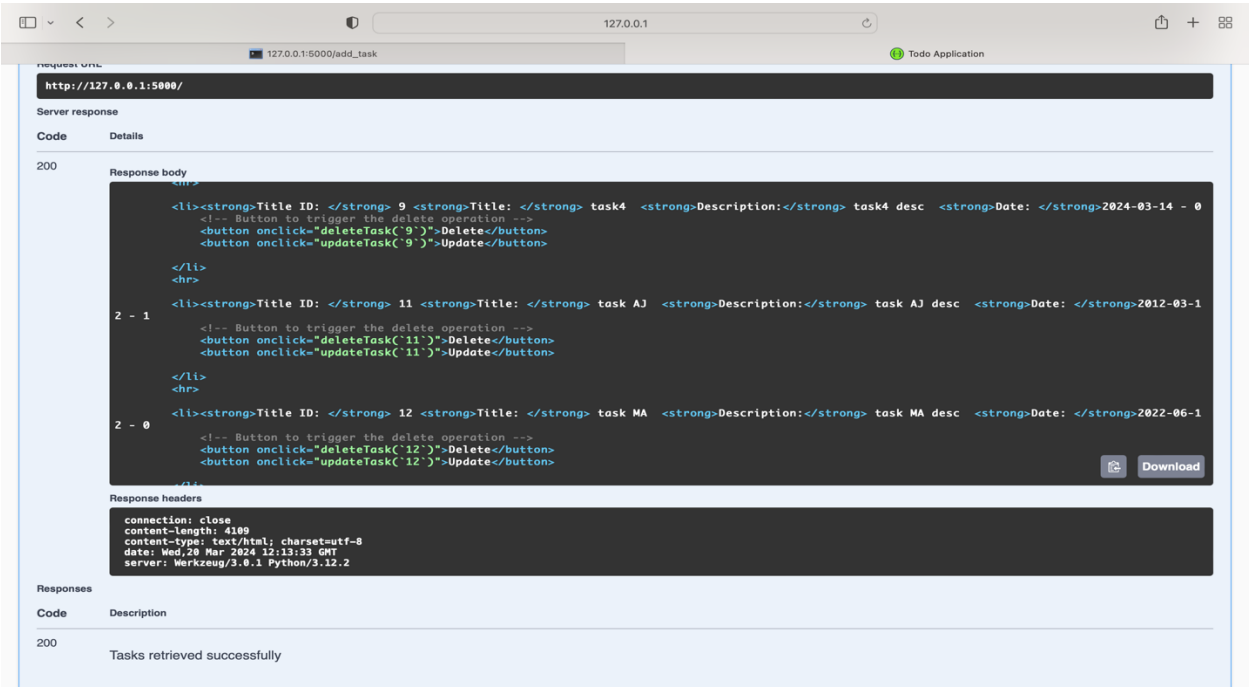


NAME: SRI HARI DHEERAJ KOMMINENI
CWID: 885177196

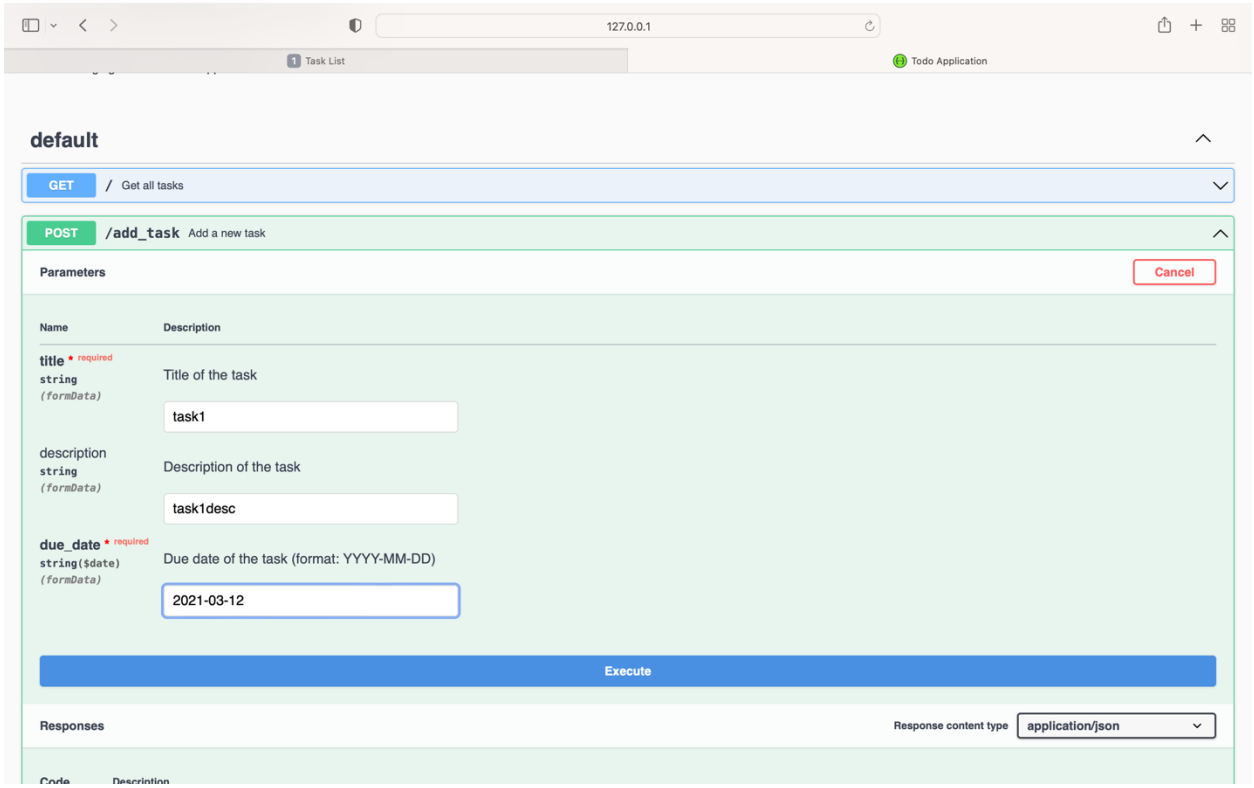
Demonstrating each API in the Swagger

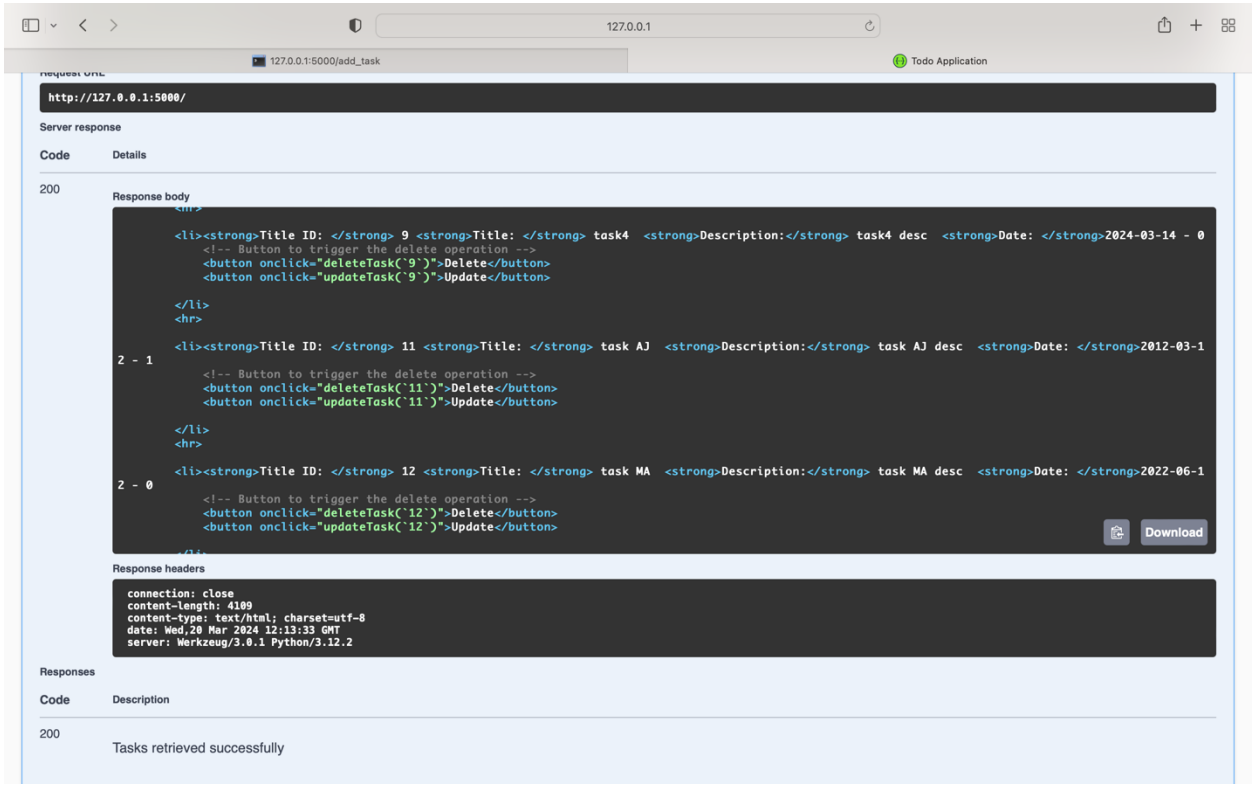
- **GET in Swagger**



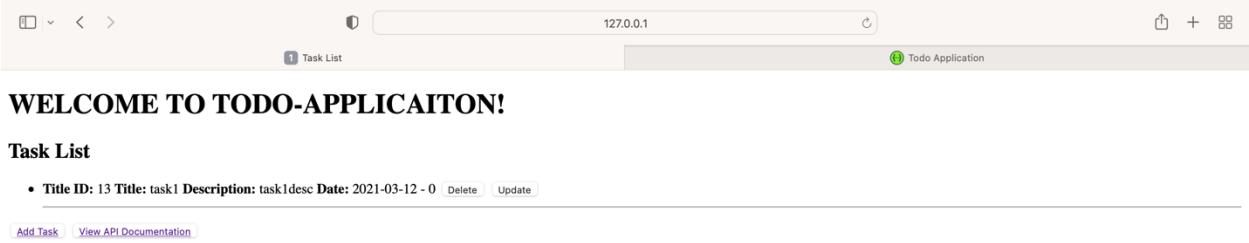


- **POST in Swagger**

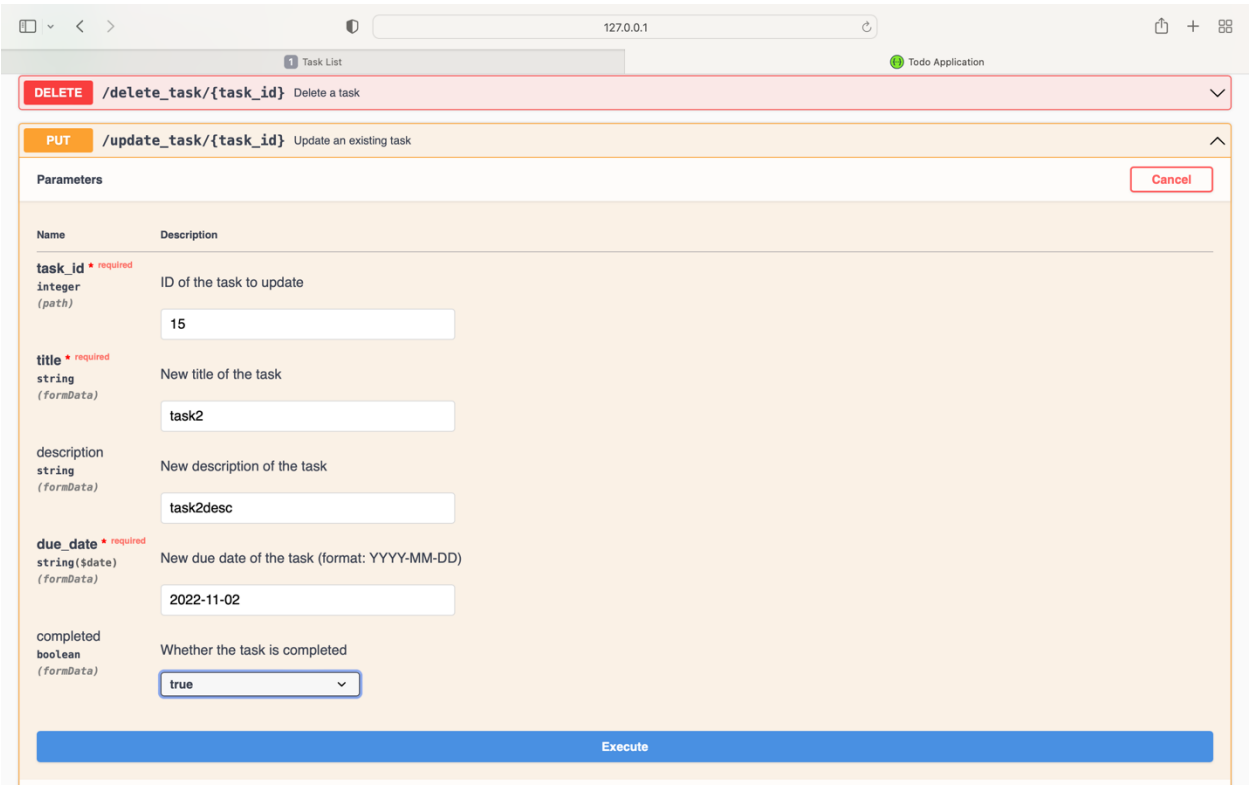




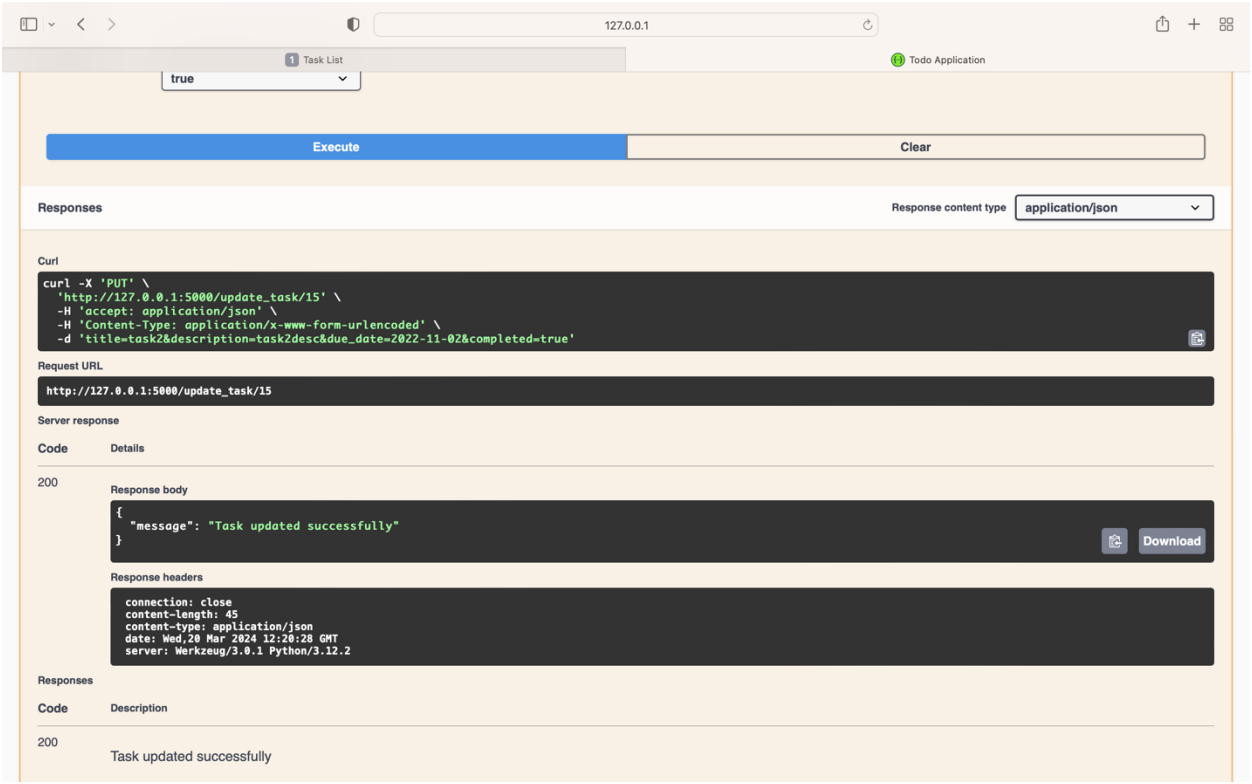
NAME: SRI HARI DHEERAJ KOMMINENI
CWID: 885177196

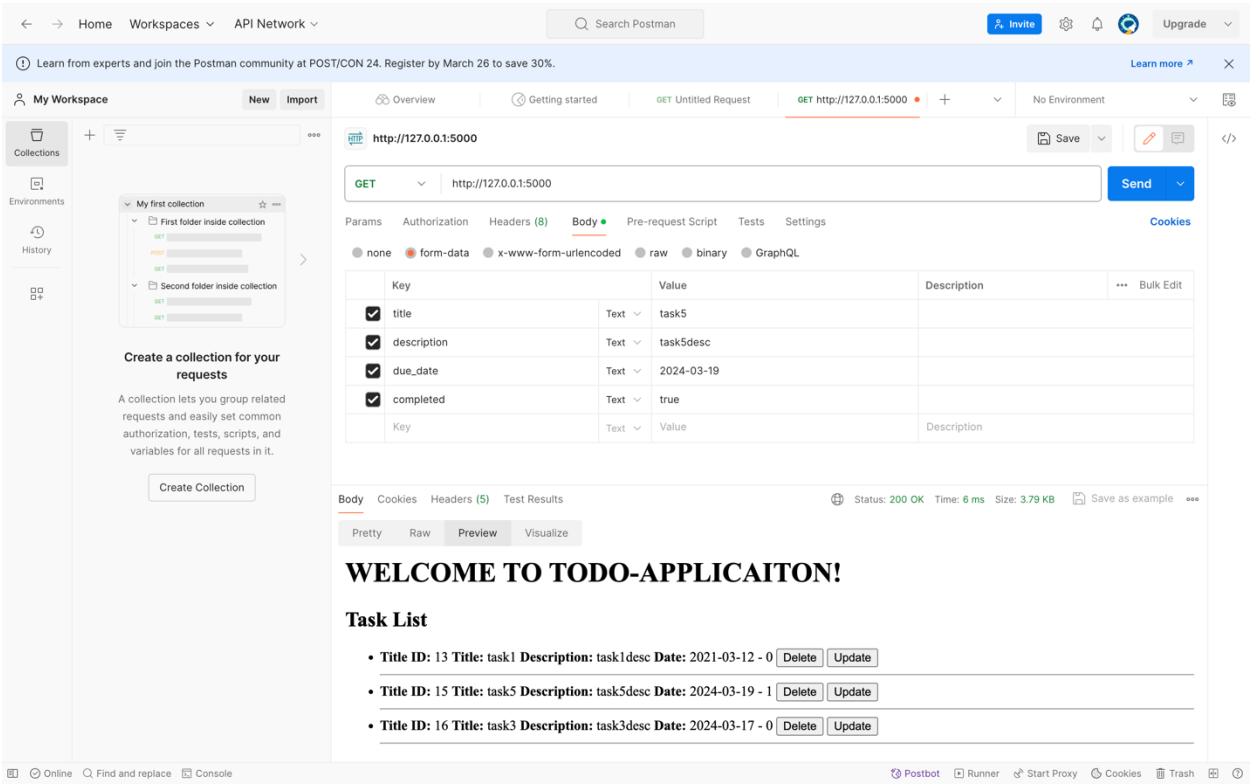


- **PUT in Swagger**

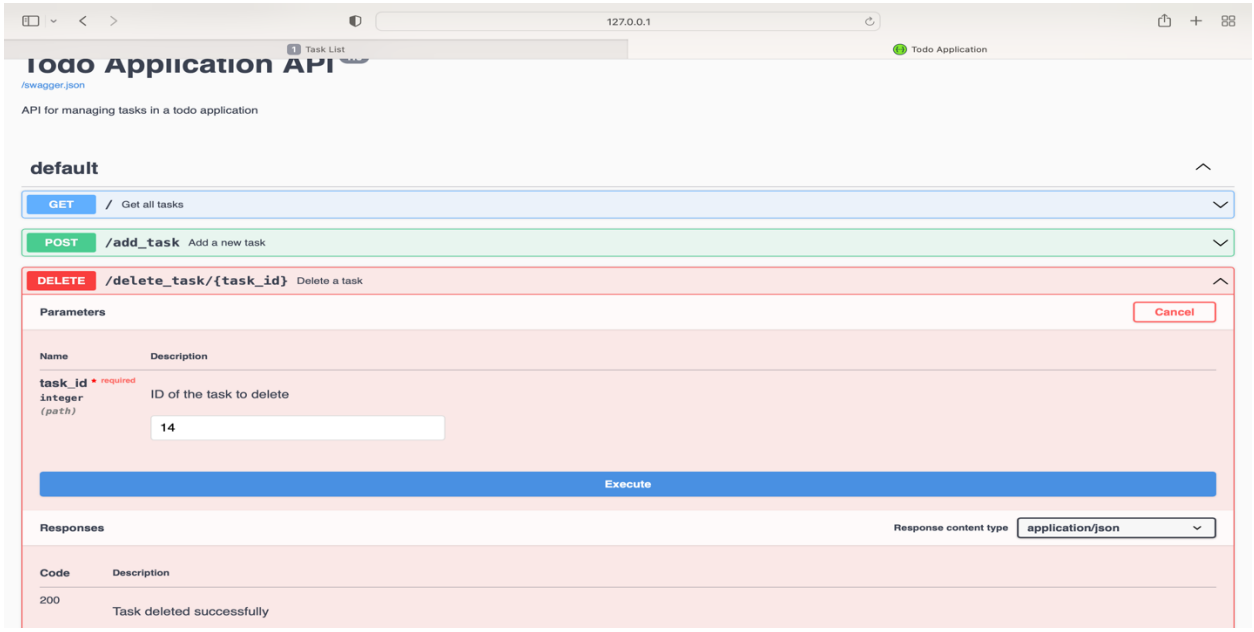


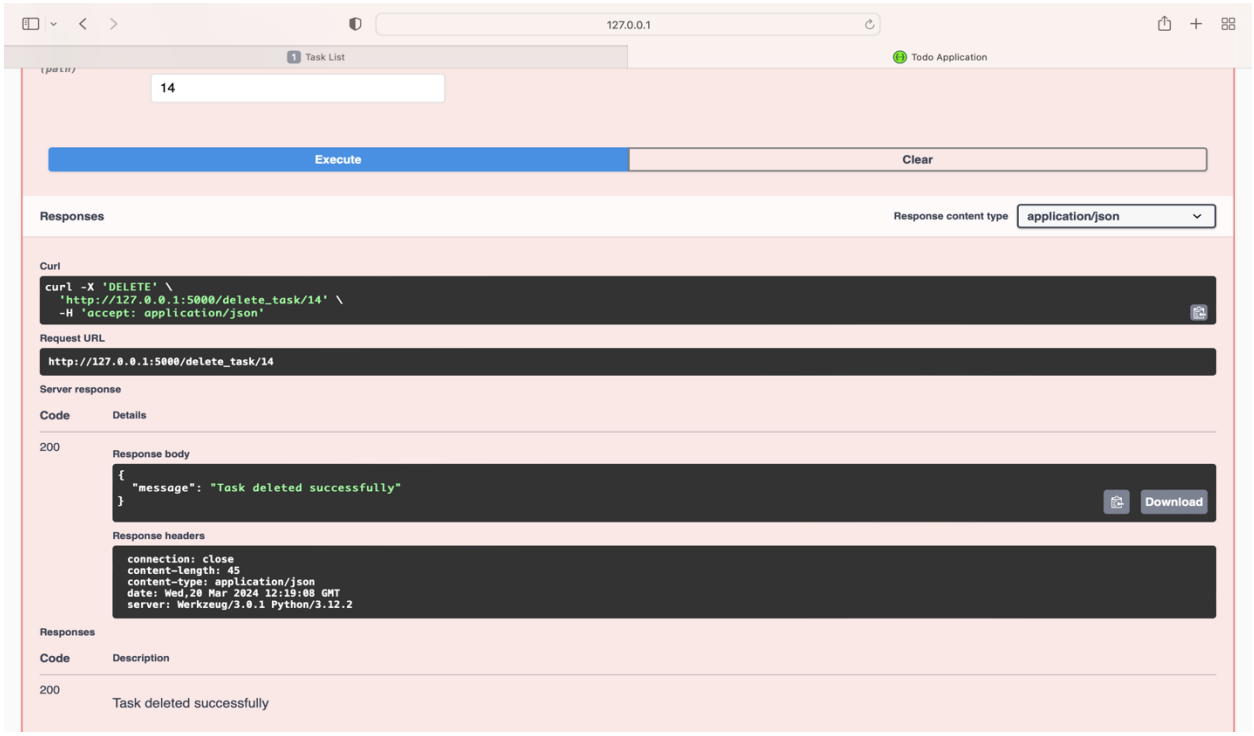
NAME: SRI HARI DHEERAJ KOMMINENI
CWID: 885177196



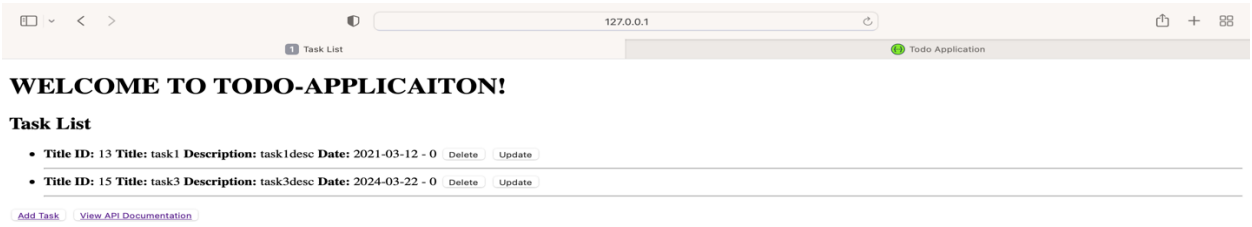


- **DELETE in Swagger**



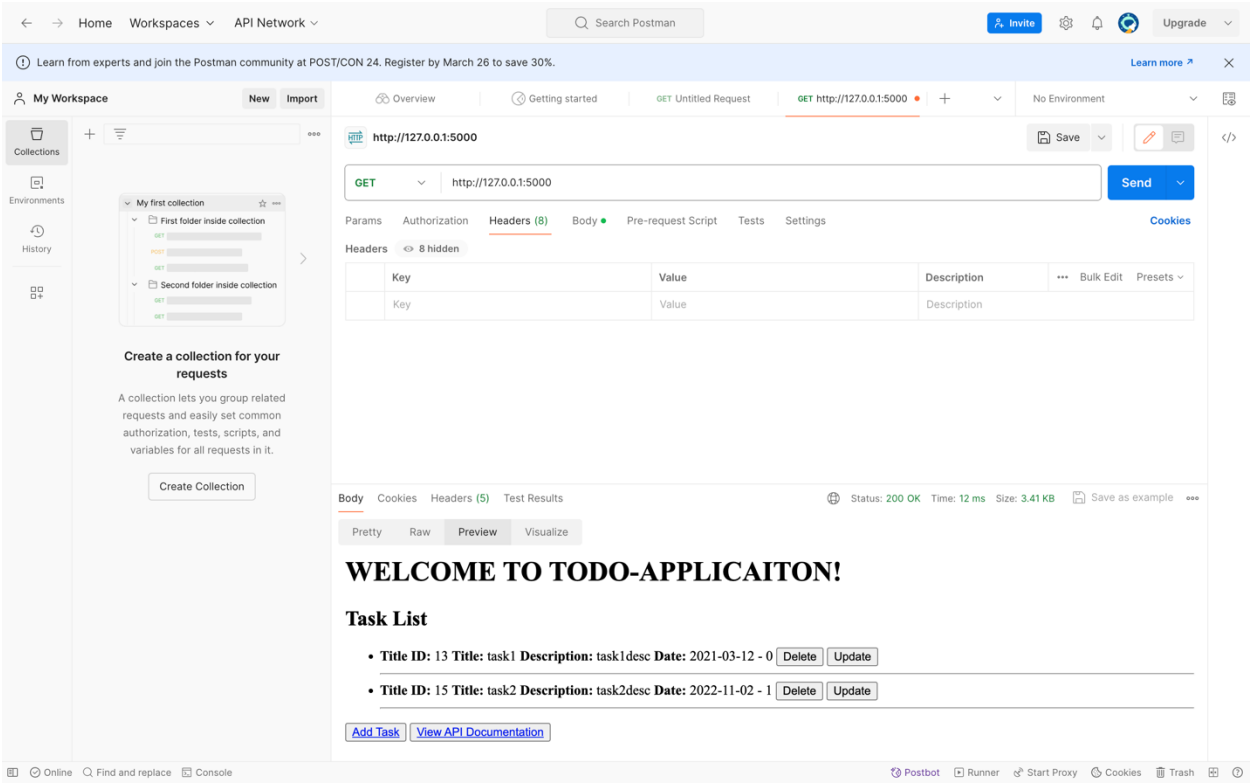


NAME: SRI HARI DHEERAJ KOMMINENI
CWID: 885177196



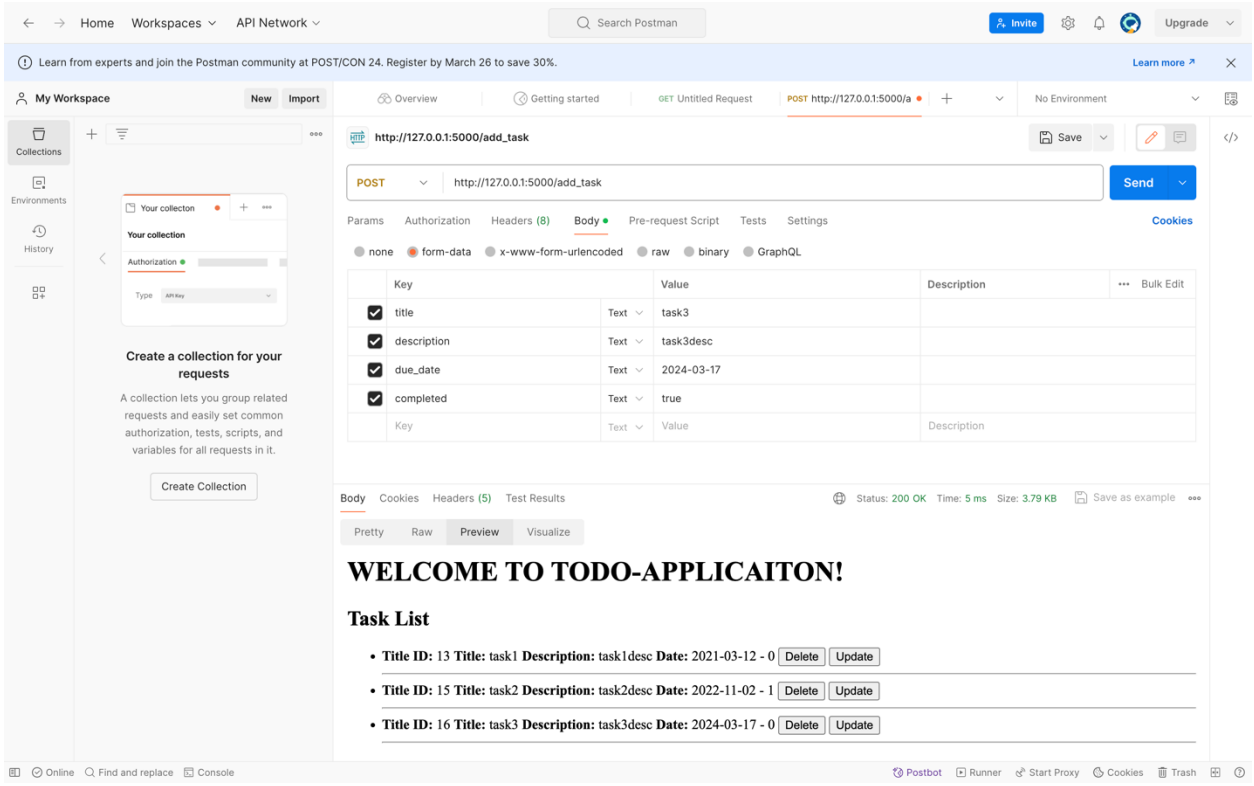
The following are the screenshots of each API request and response in POSTMAN.

- **GET in POSTMAN**



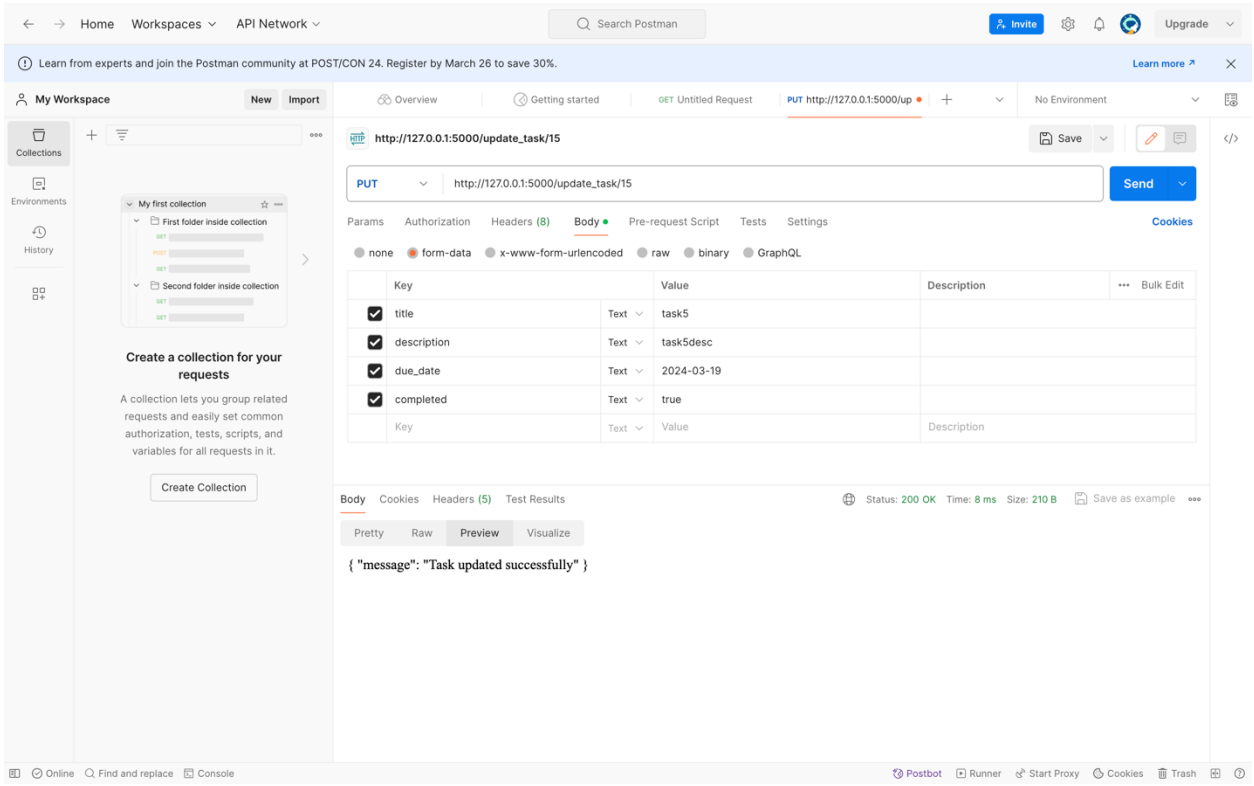
NAME: SRI HARI DHEERAJ KOMMINENI
CWID: 885177196

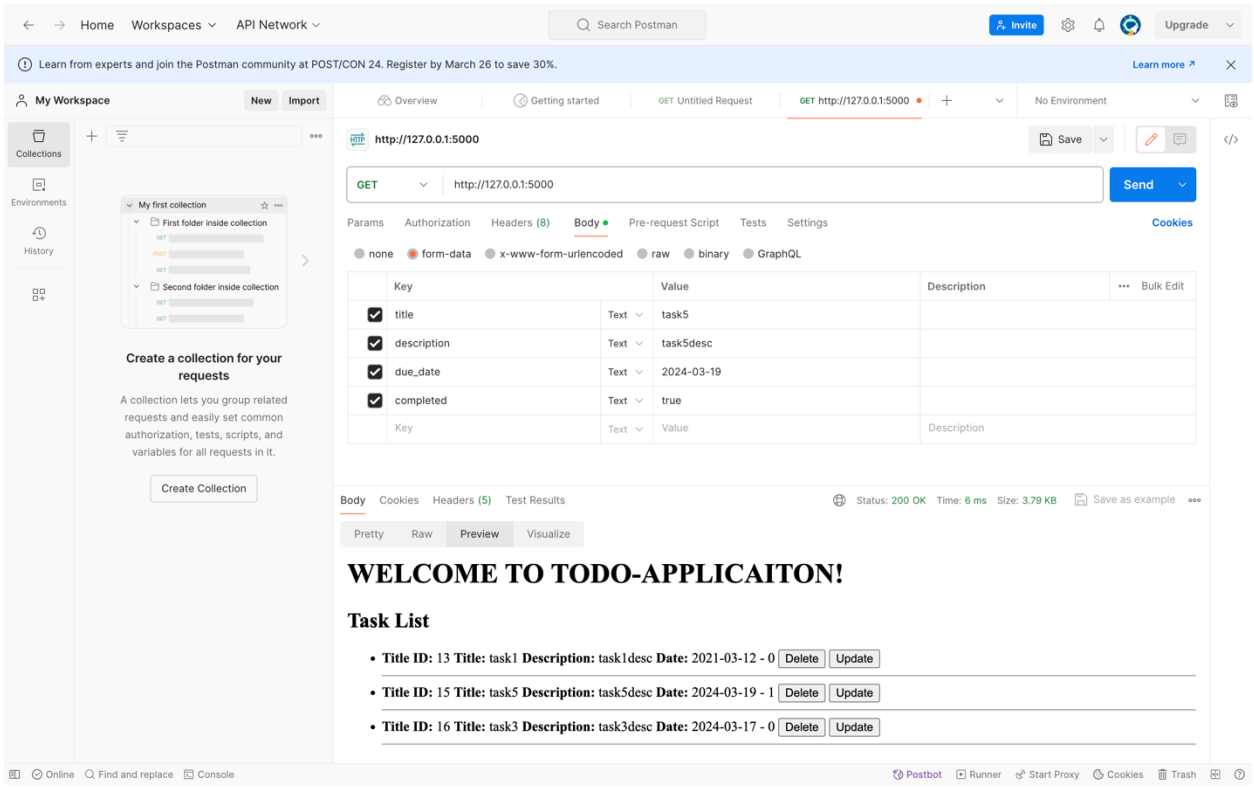
- **POST in POSTMAN**



NAME: SRI HARI DHEERAJ KOMMINENI
CWID: 885177196

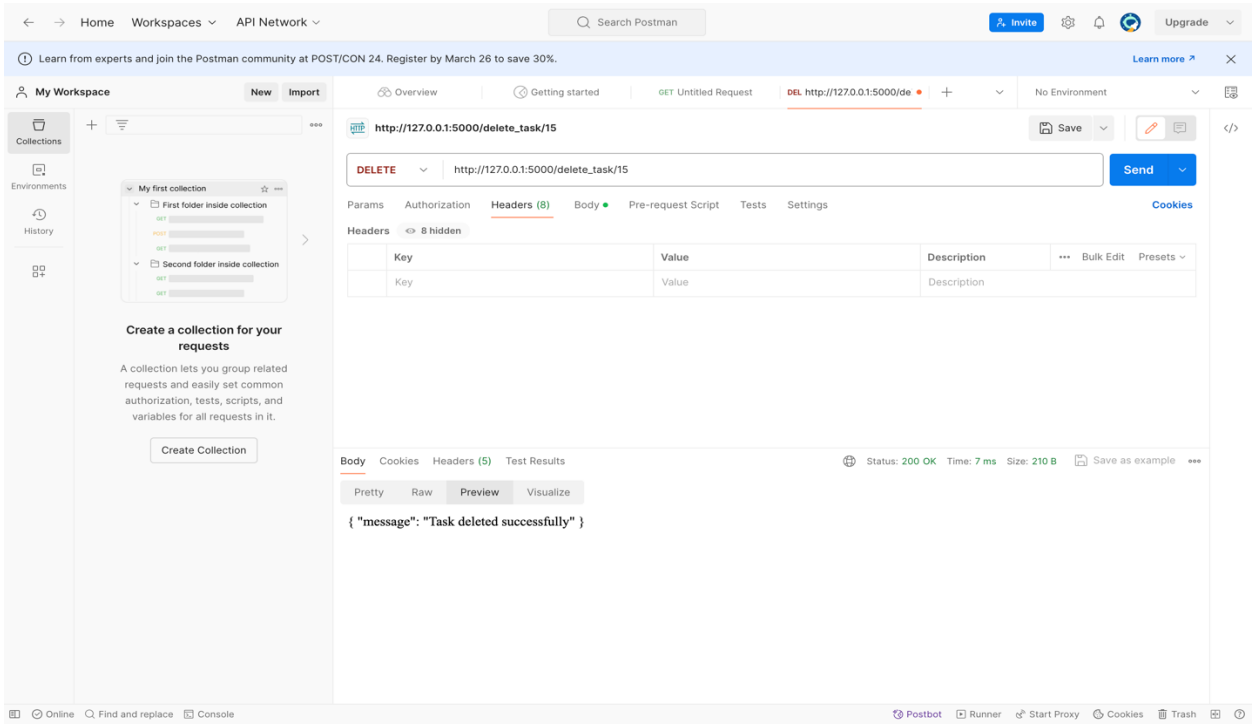
- **PUT in POSTMAN**

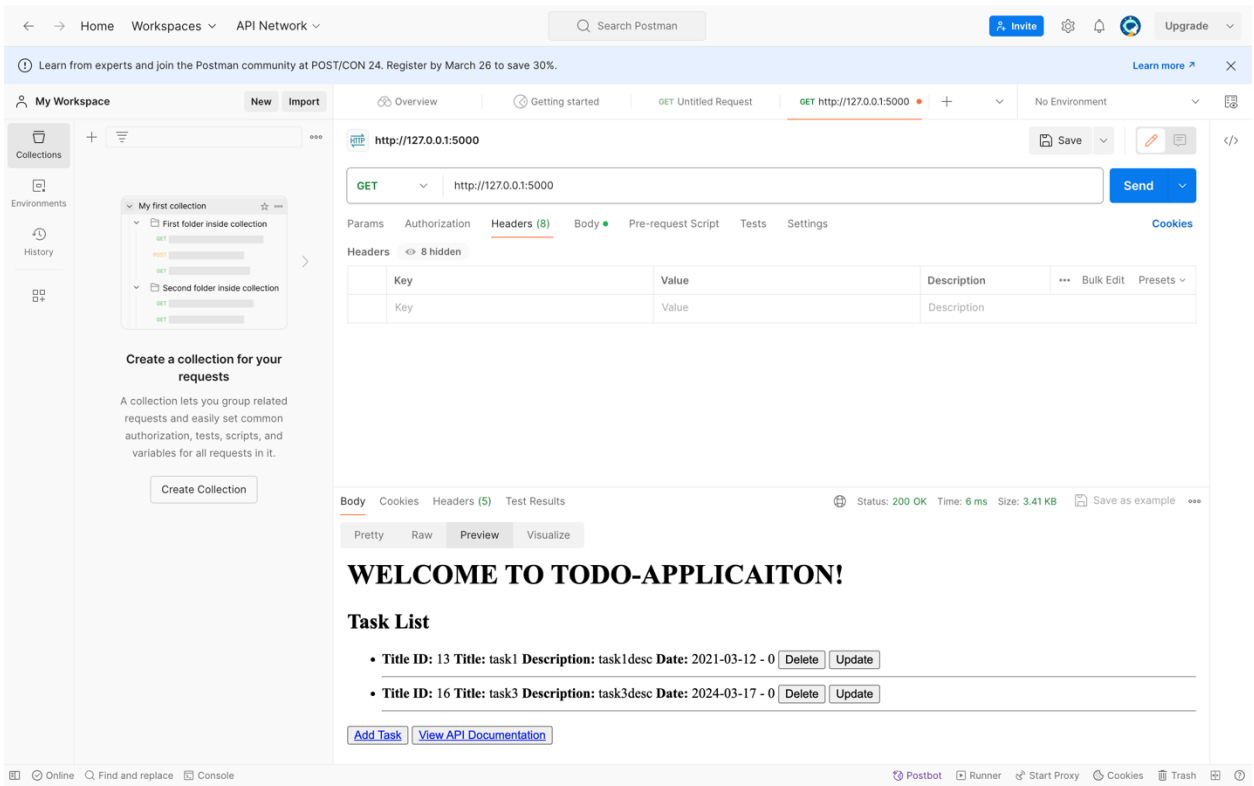




NAME: SRI HARI DHEERAJ KOMMINENI
CWID: 885177196

- **DELETE in POSTMAN**

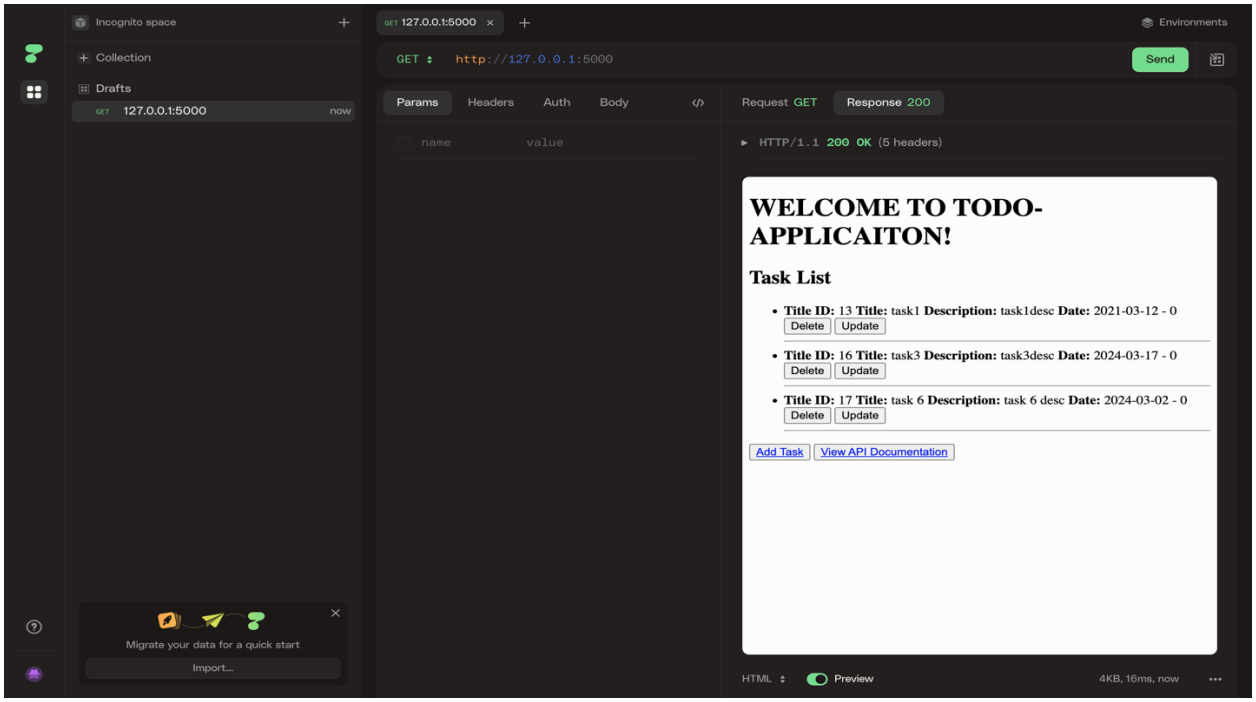




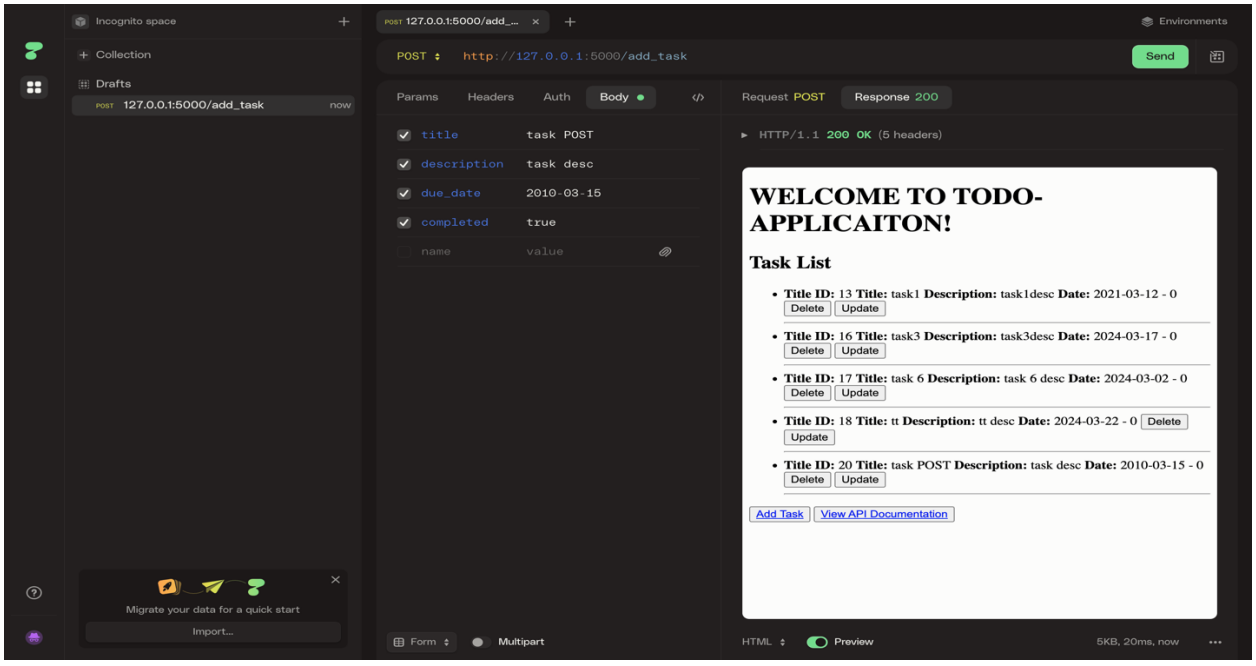
NAME: SRI HARI DHEERAJ KOMMINENI
CWID: 885177196

The following are the screenshots of each API request and response in HTTPie.

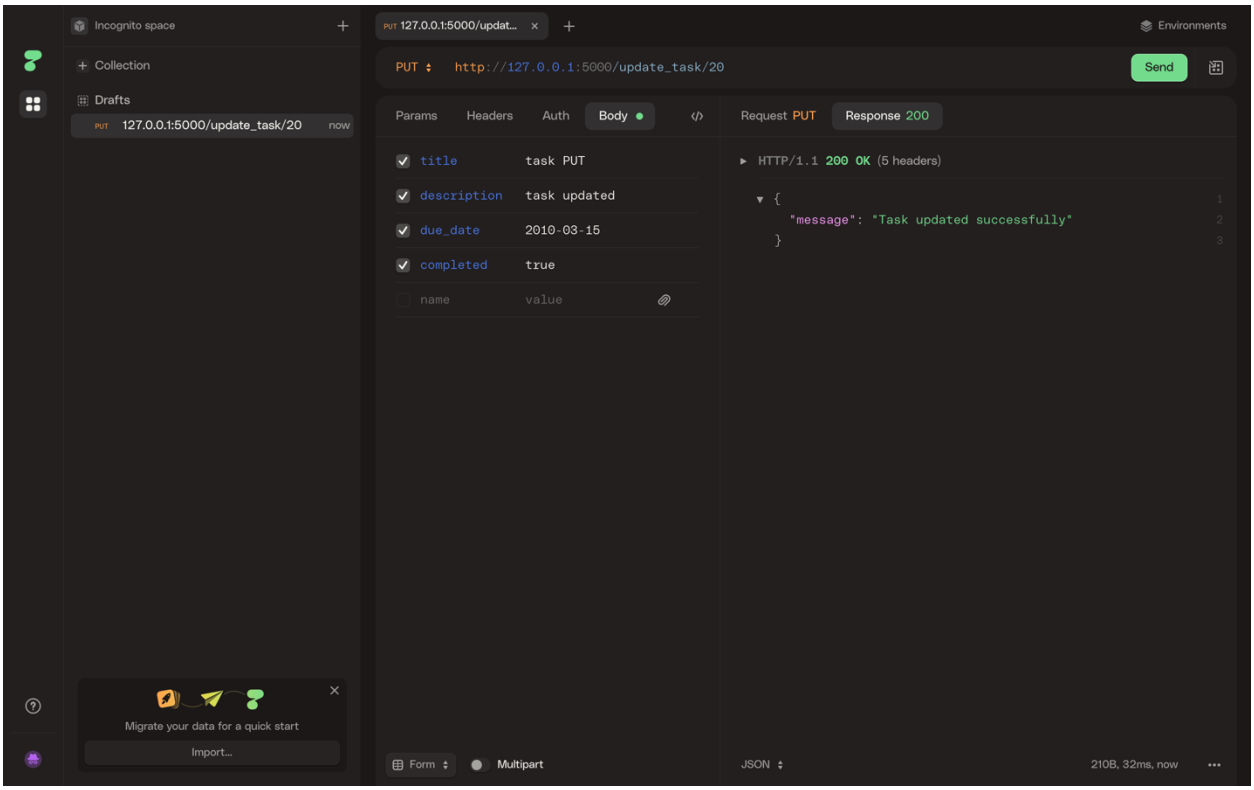
- **GET in HTTPie**

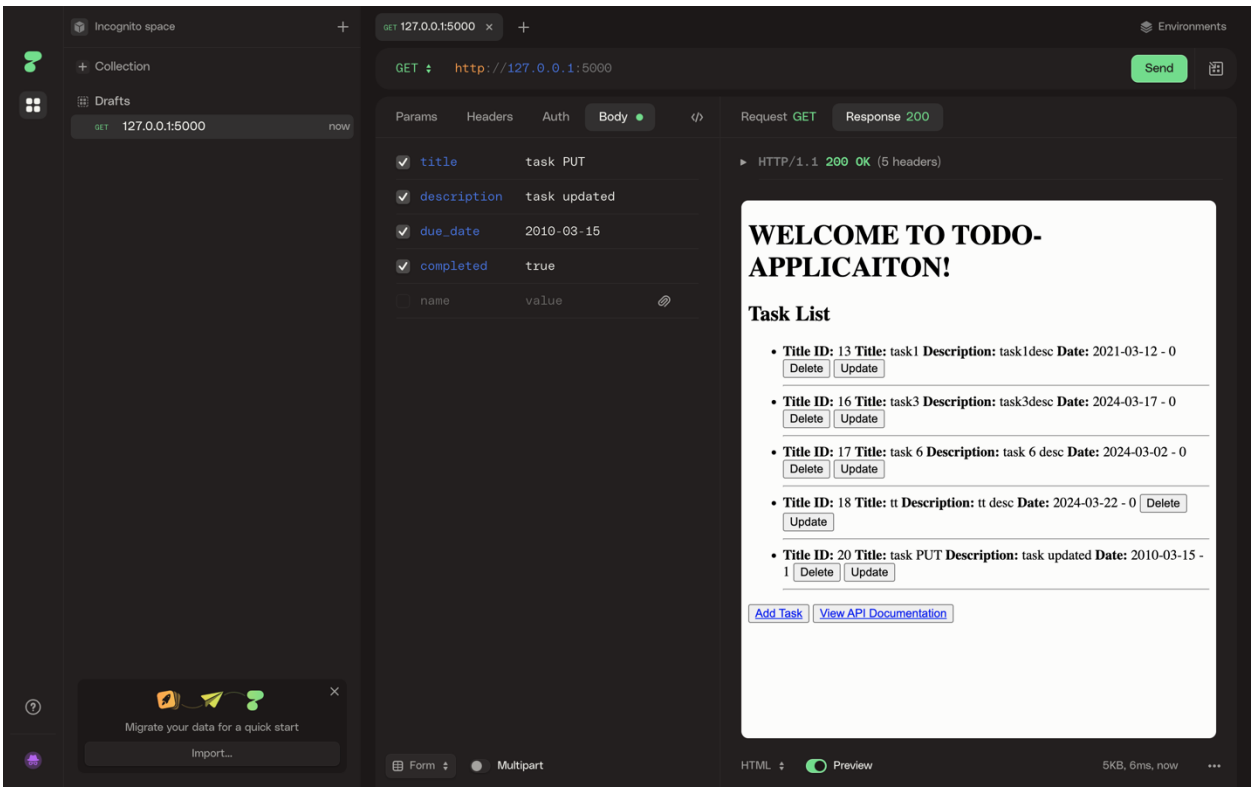


- **POST in POSTMAN**



- **PUT in HTTPie**





NAME: SRI HARI DHEERAJ KOMMINENI
CWID: 885177196

- **DELETE in HTTPie**