

UST Task Manager

Project: Task Tracker API(Phase 1)

Tech Stack: FastAPI + Python

Security: JWT authentication (HS256)

Data Storage: In-Memory Lists Only (No DB)

Goal: Build a secure backend API that manages tasks.

1. Project Overview

You will build a JWT-secured Task Tracker API with the following modules:

1. User Login API → Returns JWT token
2. Task CRUD API (Create, Read, Update, Delete) → Requires JWT
3. In-memory data storage → No database
4. Input validation using Pydantic models

This API will be used later as the backend for a frontend app.

Scenario 1

Scenario 1: Successful Login

Input

```
{  
  "username": "rahul",  
  "password": "password123"  
}
```

Output

task Manager

```
{  
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  
  "token_type": "bearer"  
}
```

Request body required

[Edit Value](#) | [Schema](#)

```
{  
  "username": "rahul",  
  "password": "password123"  
}
```

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "username": "rahul",
    "password": "password123"
}'
```



Request URL

```
http://127.0.0.1:8000/login
```

Server response

Code Details

200

Response body

```
{
  "access_token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJyWh1bCTStmV4cCI6MTc2MDQ1MzA4Mg.jy_KG-BRtPn4pZSpdF1172pIjclShE_Di89ExpoKog",
  "token_type": "Bearer"
}
```



Download

Response headers

```
content-length: 165
content-type: application/json
```

✖ Scenario 2: Login Failure

Input

```
{  
    "username": "rahul",  
    "password": "wrongpass"  
}
```

Output

```
{"detail": "Invalid username or password"}
```

Request body required

[Edit Value](#) | [Schema](#)

```
{  
    "username": "rahul",  
    "password": "wrongpass"  
}
```

Responses

Curl

```
curl -X 'POST' \  
  'http://127.0.0.1:8000/login' \  
  -H 'accept: application/json' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "username": "rahul",  
    "password": "wrongpass"  
  }'
```

Request URL

```
http://127.0.0.1:8000/login
```

Server response

Code	Details
403	Error: Forbidden

Undocumented

Response body

```
{  
    "detail": "Invalid Credentials"  
}
```

Scenario 3: Create Task

Input

```
{  
  "title": "Book train tickets",  
  "description": "Book IRCTC tickets to Chennai"  
}
```

Output

```
{  
  "id": 1,  
  "title": "Book train tickets",  
  "description": "Book IRCTC tickets to Chennai",  
  "completed": false  
}
```

Manager



```
"description": "Book IRCTC tickets to Chennai",  
"completed": false  
}
```

Request body required

[Edit Value](#) | [Schema](#)

```
{  
  "title": "Book train tickets",  
  "description": "Book IRCTC tickets to Chennai"  
}
```

Request URL

```
http://127.0.0.1:8000/tasks
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "id": 2, "title": "Book train tickets ", "description": "Book IRCTC tickets to Chennai", "completed": false }</pre> <p>Response headers</p> <pre>content-length: 102 content-type: application/json date: Tue, 25 Nov 2025 06:22:22 GMT server: unicorn</pre>

✖ Scenario 4: Create Task with Missing Title

Input

```
{  
    "description": "Something"  
}
```

Output

```
{  
    "detail": [  
        {  
            "loc": ["body", "title"],  
            "msg": "field required",  
            "type": "value_error.missing"  
        }  
    ]  
}
```

Request body required

[Edit Value](#) | [Schema](#)

```
{  
  "description": "Something"  
}
```

Request URL

<http://127.0.0.1:8000/tasks>

Server response

[Code](#) [Details](#)

422 Error: Unprocessable Content

Response body

```
{  
  "detail": [  
    {  
      "type": "missing",  
      "loc": [  
        "body",  
        "title"  
      ],  
      "msg": "Field required",  
      "input": {  
        "description": "Something"  
      }  
    }  
  ]  
}
```

✓ Scenario 5: Update Task

Input

```
{  
  "title": "Pay Water Bill",  
  "description": "Pay BWSSB bill",  
}  
  
k Manager
```

Output

```
{  
  "id": 1,  
  "title": "Pay Water Bill",  
  "description": "Pay BWSSB bill",  
  "completed": true  
}
```

Request body required

[Edit Value](#) | [Schema](#)

```
{  
  "title": "Pay Water Bill",  
  "description": "Pay BWSSB bill",  
  "completed": true  
}
```

Request URL

```
http://127.0.0.1:8000/tasks/1
```

Server response

Code	Details
------	---------

200	Response body
-----	---------------

```
{  
    "id": 1,  
    "title": "Pay Water Bill",  
    "description": "Pay BISSB bill",  
    "completed": true  
}
```

✖ Scenario 7: Delete Non-Existing Task

Response

```
{"detail": "Task not found"}
```

Request URL

```
http://127.0.0.1:8000/tasks/3
```

Server response

Code	Details
------	---------

404	Error: Not Found
-----	------------------

Undocumented

Response body

```
{  
    "detail": "Task Not Found"  
}
```

Scenario 6: Get Non-Existing Task

Response

```
{"detail": "Task not found"}
```

Request URL

```
http://127.0.0.1:8000/tasks/3
```

Server response

Code	Details
404	Error: Not Found <small>Undocumented</small>

Response body

```
{  
    "detail": "Task Not Found"  
}
```

GET ALL:

Request URL

```
http://127.0.0.1:8000/tasks
```

Server response

Code	Details
200	

Response body

```
[  
    {  
        "id": 1,  
        "title": "Pay Water Bill",  
        "description": "Pay BISB bill",  
        "completed": true  
    }  
]
```