

"Django Todo app where users can sign up to create and manage simple daily tasks."

Name

Abubakr lqbal

Back-end Developer **Tools**

Visual Studio

Role

Django, Rest Apis

Framework

MySQL **Testing**

Relational Database

Pytest, Performance Test(Locust)

Repository

Git, Github

Index

Pytest..... Performance Test..... GitHub..... Intro

Using python 3.12 version developed todo app where users can create, delete, update or

me build **ER diagram**,

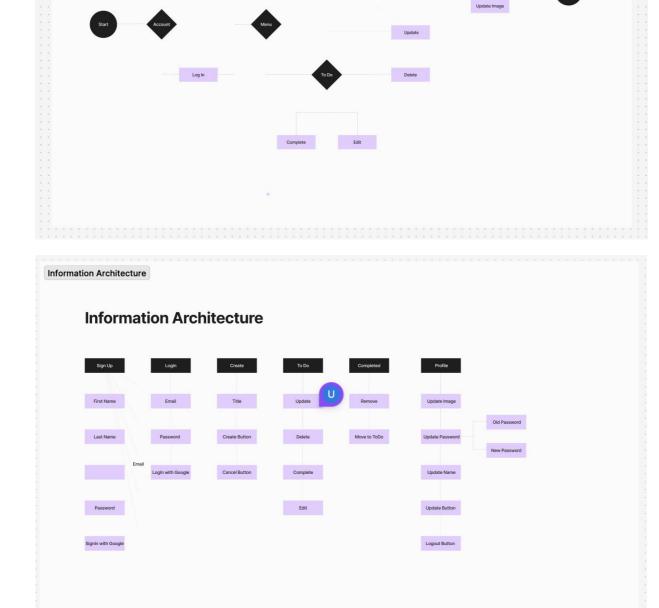
Model

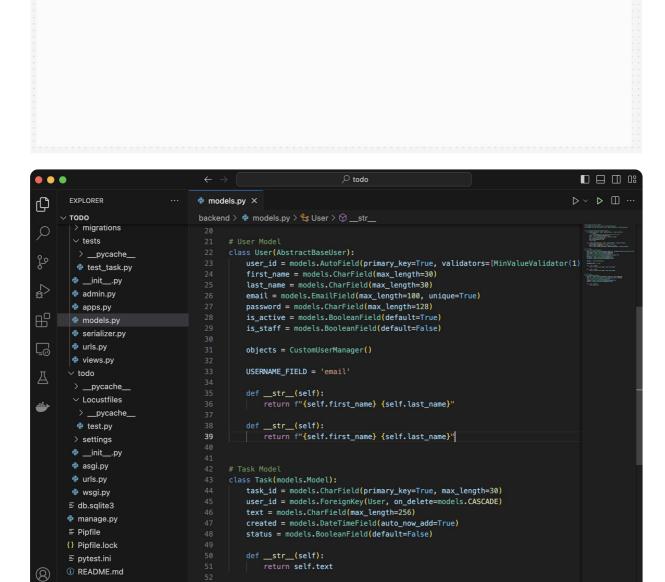
complete simple tasks

User Flow **USER FLOW**

Designed app user flow on each stage of app and information architecture which help

ER Diagram





Profile

First Name Last Name Task

Ln 39, Col 53 Spaces: 4 UTF-8 LF (Python 3.12.0 ('todo-ermTT8IX': Pipenv) @ Go Live

▷ ~ ▷ □ …

∑ zsh + ∨ □ 🛍 ··· ^ ×

Ш

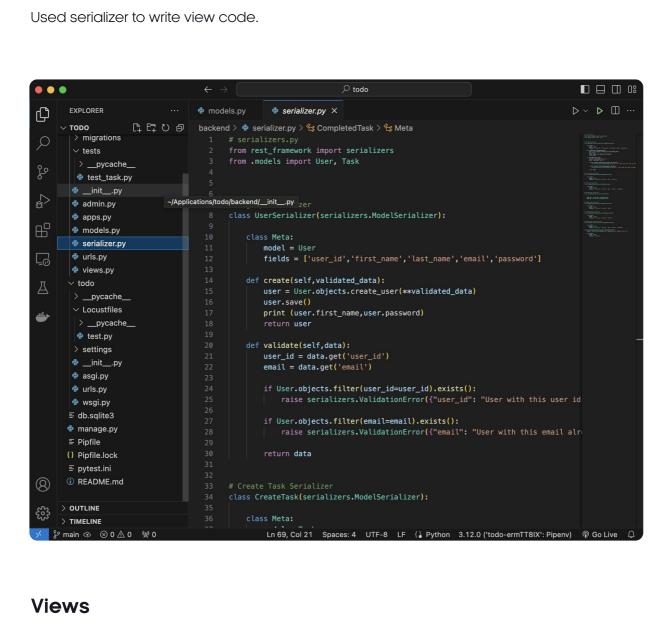
23.36 23.36

440

450

User ID

Date Or Time Created



Create, Update, View Task and Complate Task Views using Restful Apis

#signup request

def signup(request):

#login request
@api_view(['POST'])

def login(request):

views.py ×

if request.method == 'POST':

if serializer.is_valid(): serializer.save()

email = request.data.get('email')

except User.DoesNotExist:

if not user.is_active:

serializer.py

password = request.data.get('password')

user = User.objects.get(email=email)

if email is None or password is None:

 ${\tt from\ rest_framework.decorators\ import\ api_view}$

∠ todo

serializer = UserSerializer(data=request.data)

from .models import Task, User
from .serializer import UserSerializer,CreateTask, DeleteTask, ViewTask, UpdateTa

return Response(serializer.data, status=status.HTTP_201_CREATED)

return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)

if not user.check_password(password):
 return Response({"error":"Invalid email or password"}, status=status.HTTP

views.py ×

serializer = ViewTask(task, many=True).data
print(serializer)

serializer = DeleteTask(data=request.data)

user_id = serializer.validated_data['user_id']

task_id = serializer.validated_data['task_id']

return Response(serializer, status=status.HTTP_200_0K)

return Response({"status":"Failed"}, status=status.HTTP_400_BAD_REQUEST)

Ln 6, Col 1 Spaces: 4 UTF-8 LF (Python 3.12.0 ('todo-ermTT8IX': Pipenv) @ Go Live

• • • TODO

> __pycache__

dest_task.py __init__.py admin.py

apps.py

urls.py

🕏 views.py

models.py

serializer.py

> __pycache_

V Locustfiles > __pycache__

test.py > settings

__init__.py

asgi.py

urls.py

wsgi.py ≡ db.sqlite3 manage.py ≣ Pipfile

{} Pipfile.lock

≣ pytest.ini ① README.md

> TIMELINE

EXPLORER

> migrations ∨ tests

serializer.py urls.py

> __pycache__ ∨ Locustfiles

> __pycache__

asgi.py urls.py wsgi.py ■ db.sqlite3

manage.py

{} Pipfile.lock

≡ pytest.ini

README.md

≣ Pipfile

∨ todo

ဖို့ main နာ ⊗ 0 <u>Λ</u> 0 (№) 0

Д

中になり

> OUTLINE > TIMELINE \$° main ⊕ ⊗ 0 △ 0 № 0

Serializers

@api_view(['POST']) def create(request): > __pycache__ if request.method == 'POST':
 serializer = CreateTask(data=request.data) test_task.py __init__.py if serializer.is_valid(): admin.py serializer.save()
return Response({"Created": "Your task is created successfully"}, stareturn Response({"Error": "Task not created", "Details": serializer.error apps.py models.py

@api_view(['GET'])

def view_task(request): if request.method == 'GET':

@api_view(['DELETE'])

if request.method == 'DELETE':

def delete(request):

models.py

```
OUTLINE
                                                                    if Task.objects.filter(user_id=user_id, task_id=task_id).exists():
    Task.objects.filter(user_id=user_id, task_id=task_id).delete()
        > TIMELINE
 Ln 6, Col 1 Spaces: 4 UTF-8 LF ( Python 3.12.0 ('todo-ermTT8IX': Pipenv) @ Go Live
Urls
Worked on each each endpoint to make the url work.
                                                                                                                                                       • • •
          EXPLORER
                                             models.py
                                                                                        urls.py
                                                                  serializer.pv
                           ច្ចុះ ជា
           > migrations
                                                      from django.urls import {\bf path}
                                                      from . import views
           ∨ tests
            test_task.py
                                                      urlpatterns = [
           🕏 __init__.py
                                                          path('signup/', views.signup, name='signup'),
 ₹
                                                          path('login/', views.login, name='login'),
path('create_task/', views.create, name='create_task'),
path('view_task/',views.view_task, name='view_task'),
           admin.py
           apps.py
 models.py
                                                           path('delete/', views.delete, name='delete'),
           e serializer.py
                                                           path('update_task/<int:pk>/', views.update, name='update_task'),
path('completed/<int:pk>/', views.completed, name='completed'),
path('view_single_user_tasks/<int:pk>/tasks/', views.view_single_user_tasks,
          🕏 urls.py
           🕏 views.py
          ∨ todo
  Д
           > __pycache__
            > __pycache_
            test.py
           > settings
           __init__.py
           asgi.py
           urls.py
           wsgi.py
          ≡ db.sqlite3
          manage.py
          ≡ Pipfile
         {} Pipfile.lock
          ≡ pytest.ini

 README.md

 (8)
        > OUTLINE
        > TIMELINE
                                                                  Ln 15, Col 1 Spaces: 4 UTF-8 LF ( Python 3.12.0 ('todo-ermTT8IX': Pipenv) @ Go Live
```

Used 4 different case scenarios like successfully login, wrong password, wrong email

backend > tests > ♦ test_task.py > ⊕ test_create_task_completed

@pytest.mark.django_db

first_name = 'test last_name = 'user'

def test_login_successfully():

password = 'password123' user = User.objects.create_user(

email=email,

client = APIClient()

data = {

backend/tests/test_task.py

email = 'testuser@gmail.com'

first_name=first_name,
 last_name=last_name,

'password': password

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

response = client.post(url, data, format='json')

assert response.status_code == status.HTTP_200_0K assert response.data['success']== 'Logged in successfully' assert response.data['status']== 'Active'

platform darwin — Python 3.12.0, pytest-8.3.2, pluggy-1.5.0 django: version: 5.1, settings: todo.settings.development (from ini) rootdir: /Users/abubakr/Applications/todo configfile: pytest.ini plugins: django-4.8.0 collected 4 items

Ln 110, Col 32 Spaces: 4 UTF-8 LF (Python 3.12.0 ('todo-ermTT8IX': Pipenv)

password=password

∠ todo

test_task.py ×

{} Pipfile.lock > TIMELINE ြူ main နာ ⊗ 0 🛦 0 100 User test/10 user per second

0% failure

Script: test.py

Туре

Locust Test Report

Requests

During: 25/08/2024, 13:25:10 - 25/08/2024, 13:25:24

Target Host: http://localhost:8000

Request Statistics

Aggregated

Aggregated

Pytest

and create task.

EXPLORER

√ backend

∨ tests

> .pytest_cache

> __pycache_ > migrations

> __pycache__

admin.py

apps.py models.py

∨ todo > __pycache__

serializer.py urls.py

> __pycache__ test.py

> settings __init__.py asgi.py

urls.py

wsgi.py ≡ db.sqlite3 manage.py ≣ Pipfile

test_task.py -_init__.py

Response Time Statistics 50%ile (ms) Name 60%ile (ms) 70%ile (ms) 80%ile (ms) 90%ile (ms) 95%ile (ms) 99%ile (ms) 100%ile (ms) POST 380 /login/

Min (ms)

Max (ms)

Average size (bytes)

Average (ms)

```
Response Times (ms)
                                                              Oth percentile 95th percen
  Total Requests per Second
1000 User test/100 user per second
43% failure
                Locust Test Report
               During: 25/08/2024, 13:34:00 - 25/08/2024, 13:34:13
                Target Host: http://localhost:8000
                Request Statistics
                                                          Average (ms)
                                                                       Min (ms) Max (ms)
                                                                                            Average size (bytes)
                                                                                             3699.64
                         Aggregated
                Response Time Statistics
                                     50%ile (ms)
                                                            70%ile (ms)
                                                                        80%ile (ms)
                                                                                   90%ile (ms)
                                                                                                          99%ile (ms)
                                                                                                                      100%ile (ms)
                          Aggregated
                                                 1300
                                                                                    3500
                                                                                                                      8300
```

How to improve failed result

 Make queries more efficient Add more servers · Optimise code

https://github.com/Abubakr1010/TODO-APP

Git/Github Created a local repository using git and deployed TODO app on github. Also seperated production and development environment so i can deploy the app later on AWS or Google cloud

Link