Django Based Chat Application

Submitted By:

Dharmendra Kumar 1805210017
Rohit Kushwaha 1805210044
Arvind Kumar Rao 1900520109003

Supervisor: Prof. Manish Gaur
Co-Supervisor: Ajeet Shukla

Table of Content

- Objective
- Introduction to Django
- Architecture
- Methodology
- Tools And Technology

Objective

The objective was to create a chat application using Django framework.

Introduction

- Django is web building framework
- It is based on python programming Language
- It is a framework that can take care of common functionality for web applications

Architecture

- Django is based on MVT(Model-View-Template) Architecture

  Which is a software design pattern for a web application.

  It has 3 main parts-

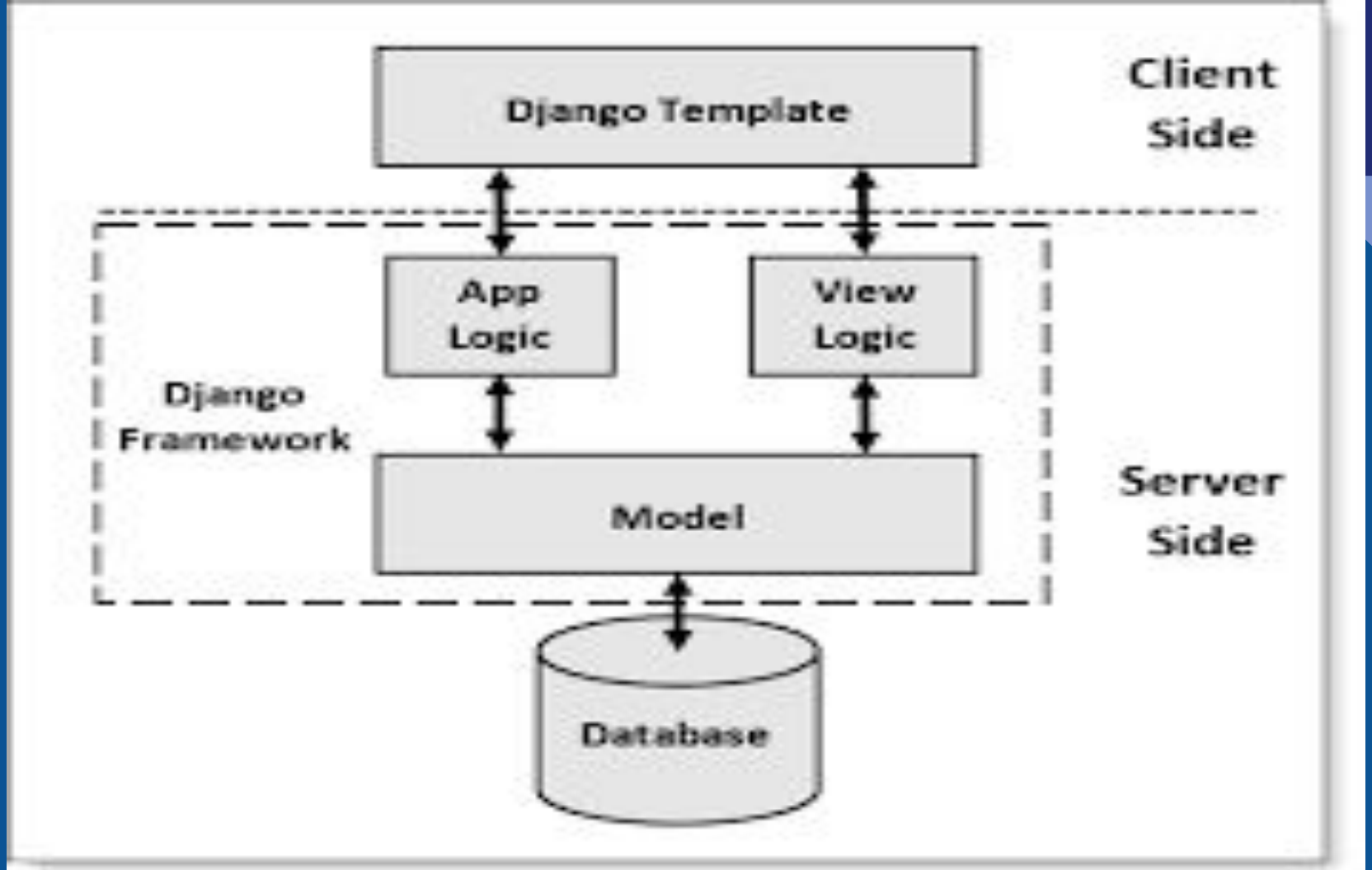
1. Model - It is the logical data structure behind the entire application and represented by a database.It act as a interface for our data.

2. View-

- A view is a function that takes a web request and return a web response.
- They usually render the the content of a web page or an error.

3. Templates-

- A template consist of static part of the desired html output
- It describes how dynamic content will be inserted.


Django Template
Client Side
App Logic
View Logic
Django Framework
Model
Server Side
Database

Methodology

- In this project we have used the channels functionality provided by Django.
- Django Channels are able to handle HTTP along with synchronous protocol like web sockets, chat bots etc.
- Channels basically allows the application to support long-running connections.

Django Channel

- The project is built on top of communication channels provided by django itself.

- Channels preserve the synchronous behaviour of Django and add a layer of synchronous protocol allowing users to write the views that are entirely synchronous, asynchronous or both.

Consumers

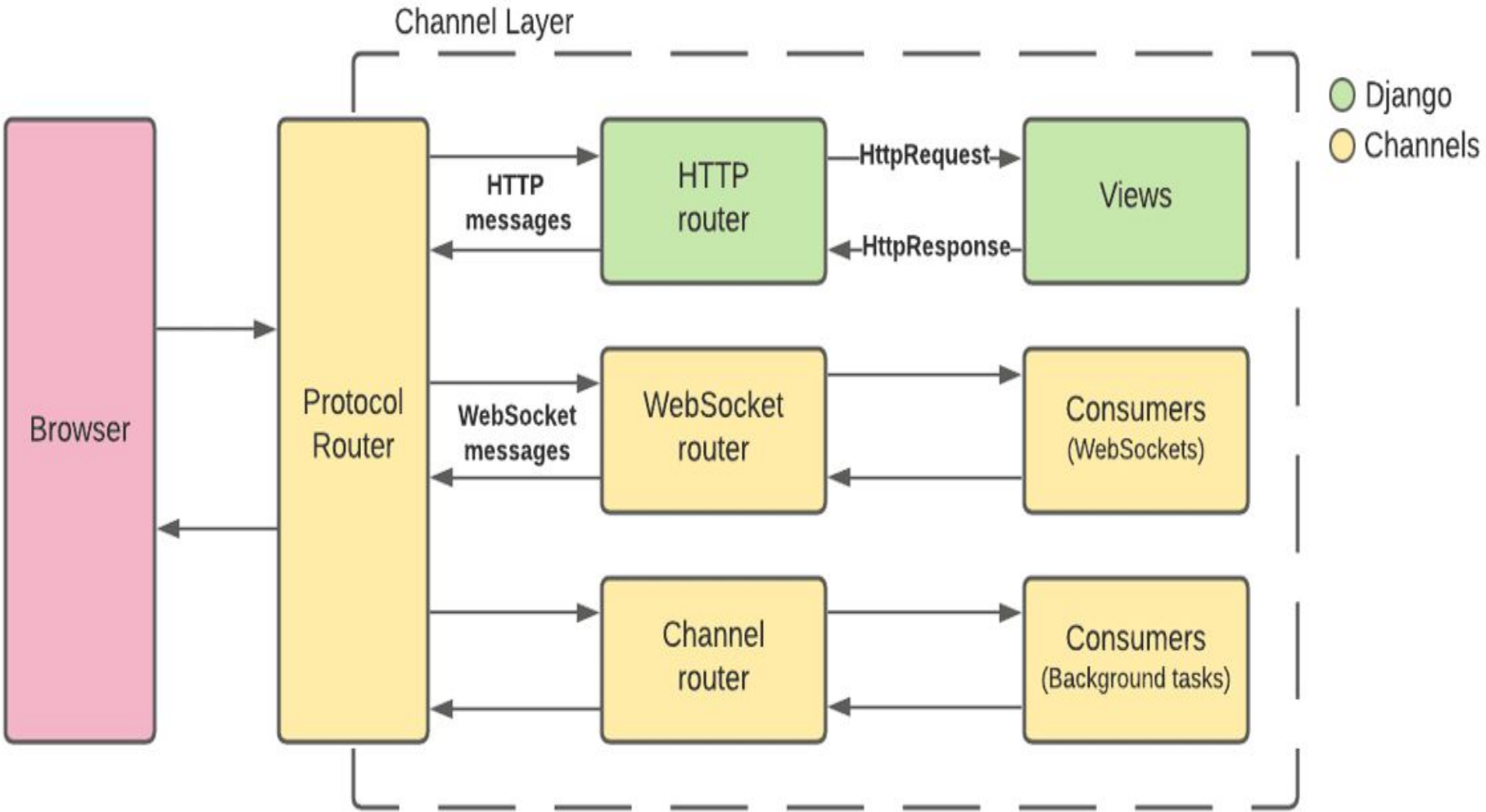
A **consumer** is a basic unit of Channels. It is an event-driven class that supports both async and sync applications. Consumers can run longer and hence they support web sockets that need persistent connection.

Consumers(continued)

1. Accept(): It accepts the incoming connections

2. Close(): It closes the current connection

3. Send(): It sends the specific message to client


Channel Layer
Django
Channels
Browser
Protocol Router
HTTP messages
HTTP router
HttpRequest
HttpResponse
Views
WebSocket messages
WebSocket router
Consumers (WebSockets)
Channel router
Consumers (Background tasks)

Tools And Technologies Used

Django- as the main web framework.

Python - as the main programming language

CSS/HTML/JavaScript - as the front end technologies.

Sqlite3 - as the database

Django Based Chat Application

Login Register

Created by

Dharmendra Kumar

Arvind Kumar Rao

Rohit Kushwaha

Sign Up

Username*

Name*

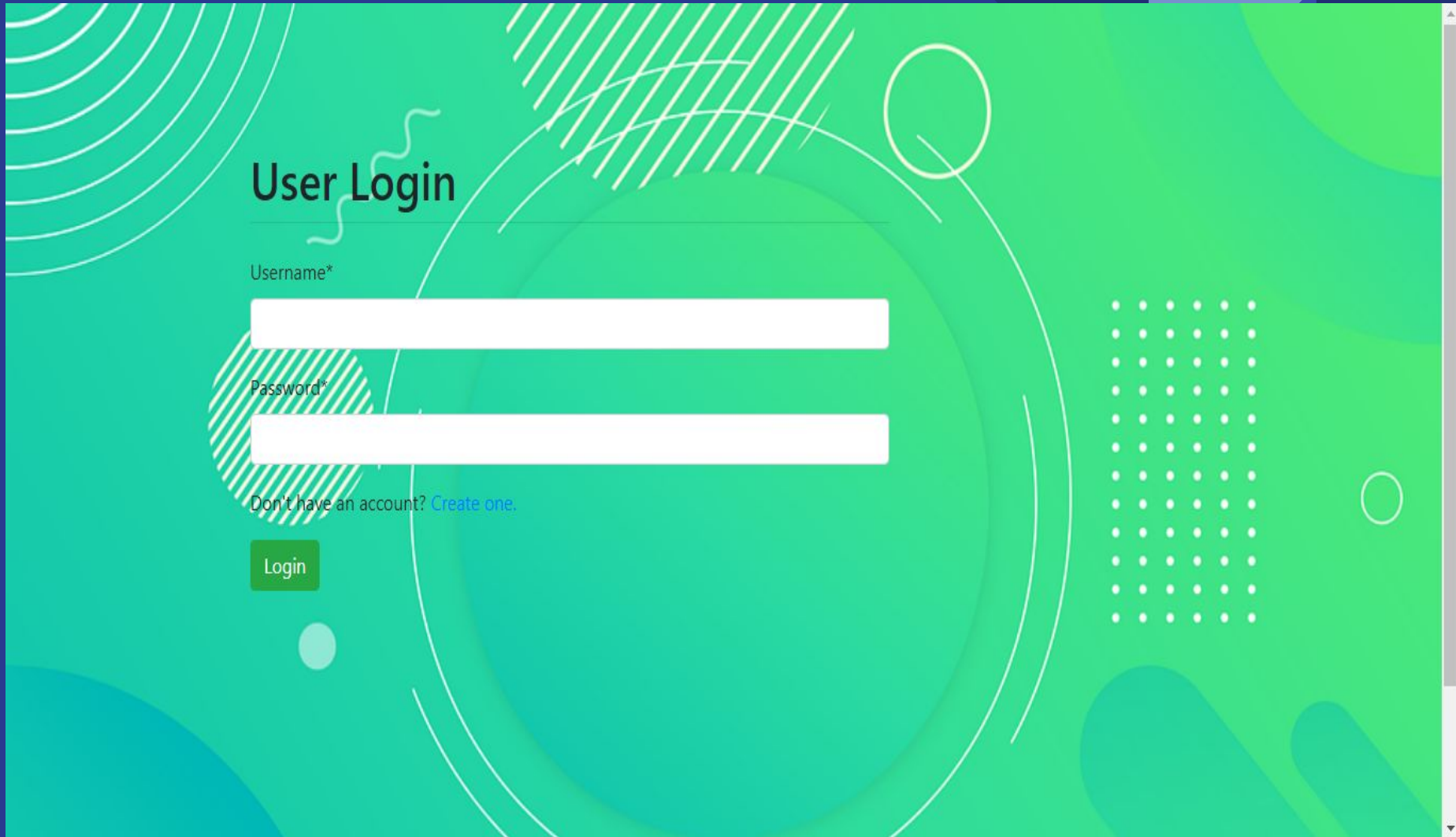
Email*

Password*

Confirm Password*

Already have an account?Login here.

Sign Up


User Login
Username*
Password*
Don't have an account? Create one.
Login

Menu

dharmendra kumar

rohit kushwaha

dharmendra kumar

Add Friends

Search by name, username

Search

dharmendra kumar

@lucifer

rohit kushwaha

@rohit

dharmendra kumar

@dharmendra17

Thank you