**1. Frontend Code (HTML, CSS)**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Responsive Webpage</title>

  <style>

    body {

      margin: 0;

      font-family: Arial, sans-serif;

    }

    .navbar {

      position: fixed;

      top: 0;

      width: 100%;

      background-color: #333;

      color: #fff;

      padding: 10px;

      text-align: center;

      z-index: 1000;

    }

    .container {

      display: flex;

      margin-top: 50px;

    }

    .left-menu {

      width: 20%;

      background-color: #f4f4f4;

      padding: 10px;

      transition: width 0.3s;

    }

    .main-content {

      flex: 1;

      padding: 10px;

    }

    .right-panel {

      width: 20%;

      background-color: #eaeaea;

      padding: 10px;

    }

    .footer {

      background-color: #333;

      color: white;

      text-align: center;

      padding: 10px;

      position: fixed;

      bottom: 0;

      width: 100%;

    }

    .collapsible {

      display: none;

    }

  </style>

</head>

<body>

  <div class="navbar">Fixed Navbar</div>

  <div class="container">

    <div class="left-menu">

      <button onclick="toggleMenu()">Toggle Menu</button>

      <div class="collapsible">Collapsible Left Menu Content</div>

    </div>

    <div class="main-content">Main Content Area</div>

    <div class="right-panel">Right Side Panel</div>

  </div>

  <div class="footer">Footer Content</div>

  <script>

    function toggleMenu() {

      const menu = document.querySelector('.collapsible');

      menu.style.display = menu.style.display === 'block' ? 'none' : 'block';

    }

  </script>

</body>

</html>

**2. Django Chat Application**

**a. Models**

from django.db import models

from django.contrib.auth.models import User

class Message(models.Model):

sender = models.ForeignKey(User, on\_delete=models.CASCADE, related\_name='sent\_messages')

receiver = models.ForeignKey(User, on\_delete=models.CASCADE, related\_name='received\_messages')

content = models.TextField()

timestamp = models.DateTimeField(auto\_now\_add=True)

def \_\_str\_\_(self):

return f"{self.sender} to {self.receiver} at {self.timestamp}"

**b. WebSocket Consumer**

from channels.generic.websocket import AsyncWebsocketConsumer

import json

class ChatConsumer(AsyncWebsocketConsumer):

async def connect(self):

self.room\_name = self.scope['url\_route']['kwargs']['room\_name']

self.room\_group\_name = f'chat\_{self.room\_name}'

await self.channel\_layer.group\_add(self.room\_group\_name, self.channel\_name)

await self.accept()

async def disconnect(self, close\_code):

await self.channel\_layer.group\_discard(self.room\_group\_name, self.channel\_name)

async def receive(self, text\_data):

data = json.loads(text\_data)

message = data['message']

await self.channel\_layer.group\_send(

self.room\_group\_name,

{'type': 'chat\_message', 'message': message}

)

async def chat\_message(self, event):

await self.send(text\_data=json.dumps({'message': event['message']}))

**c. Routing**

from django.urls import path

from .consumers import ChatConsumer

websocket\_urlpatterns = [

path('ws/chat/<str:room\_name>/', ChatConsumer.as\_asgi()),

]

**d. Django Settings**

INSTALLED\_APPS = [

'channels',

'chat',

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

]

ASGI\_APPLICATION = 'chat\_app.asgi.application'

CHANNEL\_LAYERS = {

'default': {

'BACKEND': 'channels.layers.InMemoryChannelLayer',

},

}

**e. ASGI Configuration (asgi.py)**

python

CopyEdit

import os

from django.core.asgi import get\_asgi\_application

from channels.routing import ProtocolTypeRouter, URLRouter

from channels.auth import AuthMiddlewareStack

from chat.routing import websocket\_urlpatterns

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'chat\_app.settings')

application = ProtocolTypeRouter({

"http": get\_asgi\_application(),

"websocket": AuthMiddlewareStack(URLRouter(websocket\_urlpatterns))

})

**3. AWS Lambda Functions**

**a. Lambda Function to Add Two Numbers**

def lambda\_handler(event, context):

num1 = event.get('num1')

num2 = event.get('num2')

if num1 is None or num2 is None:

return {"error": "Provide both num1 and num2"}

return {"result": num1 + num2}

**b. Lambda Function to Upload a File to S3**

import boto3

import base64

def lambda\_handler(event, context):

s3 = boto3.client('s3')

file\_content = base64.b64decode(event['file\_content'])

file\_name = event['file\_name']

bucket\_name = 'your-bucket-name'

s3.put\_object(Bucket=bucket\_name, Key=file\_name, Body=file\_content)

return {"message": f"File {file\_name} uploaded successfully to {bucket\_name}"}