# Frontend Development

## Responsive Webpage

### HTML

```html

<!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>

<link rel="stylesheet" href="styles.css">

</head>

<body>

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

<div class="content">

<div class="left-menu">Left Menu</div>

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

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

</div>

<footer>Footer</footer>

<script src="script.js"></script>

</body>

</html>

```

### CSS

```css

body {

margin: 0;

font-family: Arial, sans-serif;

display: flex;

flex-direction: column;

height: 100vh;

}

.navbar {

position: fixed;

top: 0;

width: 100%;

background-color: #333;

color: white;

text-align: center;

padding: 1rem;

}

.content {

display: flex;

margin-top: 4rem;

flex-grow: 1;

}

.left-menu {

width: 20%;

background-color: #f4f4f4;

padding: 1rem;

}

.main-content {

flex-grow: 1;

padding: 1rem;

background-color: #fff;

}

.right-panel {

width: 20%;

background-color: #f4f4f4;

padding: 1rem;

}

footer {

background-color: #333;

color: white;

text-align: center;

padding: 1rem;

}

```

### JavaScript

```javascript

window.addEventListener('resize', () => {

const width = window.innerWidth;

const body = document.body;

if (width <= 600) {

body.style.transform = 'scale(0.5)';

} else if (width <= 700) {

body.style.transform = 'scale(0.75)';

} else if (width <= 767) {

body.style.transform = 'scale(0.8)';

} else if (width <= 1600) {

body.style.transform = 'scale(0.9)';

} else {

body.style.transform = 'scale(1)';

}

});

```

# Django Chat Application

### Models

```python

from django.contrib.auth.models import User

from django.db import models

class Message(models.Model):

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

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

content = models.TextField()

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

```

### Views

```python

from django.shortcuts import render

from django.contrib.auth.decorators import login\_required

from .models import Message

from django.contrib.auth.models import User

@login\_required

def chat(request):

users = User.objects.exclude(id=request.user.id)

return render(request, 'chat.html', {'users': users})

```

### WebSocket Consumer

```python

from channels.generic.websocket import AsyncWebsocketConsumer

import json

from .models import Message

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):

message = event['message']

await self.send(text\_data=json.dumps({

'message': message

}))

```

# AWS Lambda

### Add Two Numbers

```python

def lambda\_handler(event, context):

num1 = event['num1']

num2 = event['num2']

return {

'statusCode': 200,

'body': num1 + num2

}

```

### Upload to S3

```python

import boto3

def lambda\_handler(event, context):

s3 = boto3.client('s3')

file\_content = event['file\_content']

bucket\_name = 'your-bucket-name'

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

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

return {

'statusCode': 200,

'body': 'File uploaded successfully'

}# Frontend Development

## Responsive Webpage

### HTML

```html

<!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>

<link rel="stylesheet" href="styles.css">

</head>

<body>

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

<div class="content">

<div class="left-menu">Left Menu</div>

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

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

</div>

<footer>Footer</footer>

<script src="script.js"></script>

</body>

</html>

```

### CSS

```css

body {

margin: 0;

font-family: Arial, sans-serif;

display: flex;

flex-direction: column;

height: 100vh;

}

.navbar {

position: fixed;

top: 0;

width: 100%;

background-color: #333;

color: white;

text-align: center;

padding: 1rem;

}

.content {

display: flex;

margin-top: 4rem;

flex-grow: 1;

}

.left-menu {

width: 20%;

background-color: #f4f4f4;

padding: 1rem;

}

.main-content {

flex-grow: 1;

padding: 1rem;

background-color: #fff;

}

.right-panel {

width: 20%;

background-color: #f4f4f4;

padding: 1rem;

}

footer {

background-color: #333;

color: white;

text-align: center;

padding: 1rem;

}

```

### JavaScript

```javascript

window.addEventListener('resize', () => {

const width = window.innerWidth;

const body = document.body;

if (width <= 600) {

body.style.transform = 'scale(0.5)';

} else if (width <= 700) {

body.style.transform = 'scale(0.75)';

} else if (width <= 767) {

body.style.transform = 'scale(0.8)';

} else if (width <= 1600) {

body.style.transform = 'scale(0.9)';

} else {

body.style.transform = 'scale(1)';

}

});

```

# Django Chat Application

### Models

```python

from django.contrib.auth.models import User

from django.db import models

class Message(models.Model):

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

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

content = models.TextField()

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

```

### Views

```python

from django.shortcuts import render

from django.contrib.auth.decorators import login\_required

from .models import Message

from django.contrib.auth.models import User

@login\_required

def chat(request):

users = User.objects.exclude(id=request.user.id)

return render(request, 'chat.html', {'users': users})

```

### WebSocket Consumer

```python

from channels.generic.websocket import AsyncWebsocketConsumer

import json

from .models import Message

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):

message = event['message']

await self.send(text\_data=json.dumps({

'message': message

}))

```

# AWS Lambda

### Add Two Numbers

```python

def lambda\_handler(event, context):

num1 = event['num1']

num2 = event['num2']

return {

'statusCode': 200,

'body': num1 + num2

}

```

### Upload to S3

```python

import boto3

def lambda\_handler(event, context):

s3 = boto3.client('s3')

file\_content = event['file\_content']

bucket\_name = 'your-bucket-name'

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

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

return {

'statusCode': 200,

'body': 'File uploaded successfully'

}