# THREE TIER WEB APPLICATION USING AWS SERVICES

**Goal**: A public-facing website where users can view and search for café menu items.  
**Architecture**: Three-tier (Frontend + Backend + Database)  
**AWS Services**: S3, Lambda, API Gateway, VPC.

**STEPS INVOLVED**

**Step1: Create a VPC**

**Step 2: Build the Frontend (S3)**

**Step 3: Create a backend API**

**Step 4: Connect Frontend and Backend**

**Step 1: Creating a custom VPC**

Search for VPC

A screenshot of a computer

AI-generated content may be incorrect.

Create new VPC

A screenshot of a computer

AI-generated content may be incorrect.

VPC and more gives us more precise view of the subnets and Ip addresses. Give a name to the VPC, I have given a name as My-VPC-Project1.

A screenshot of a computer

AI-generated content may be incorrect.

The default CIDR value will be 10.0.0.0/16.

A screenshot of a computer

AI-generated content may be incorrect.

I selected 1 Availability Zone for this project, which provided 1 public subnet and up to 2 private subnets. Since I only needed 1 private subnet, I chose that. A NAT Gateway wasn’t required, so I skipped it. I did enable an S3 VPC endpoint to support secure storage and communication.A screenshot of a computer

AI-generated content may be incorrect.

The VPC creation may take up to 2 mins

A white rectangular box with a black border

AI-generated content may be incorrect.

Once done, you can check it here,

A screenshot of a computer

AI-generated content may be incorrect.

You can check all your VPCs here in the console and can filter the specific VPCs

A screenshot of a computer

AI-generated content may be incorrect.

We have to create two subnets for this project, private subnet and the public subet

A screenshot of a computer

AI-generated content may be incorrect.

- Public subnet: 10.64.0.0/24

- Private subnet: 10.63.0.0/24

A screenshot of a computer

AI-generated content may be incorrect.

***Challenge Faced***

***Problem: I was unable to give the CIDR rage which won’t overlap and existing CIDR range***

***Fix:***

***• Planned subnet layout using spreadsheet/Draw.io***

***• Used valid blocks like 10.0.64.0/24 (not 10.0.64.1/24)***

***• Checked existing CIDRs in VPC console***

***• Optional: Used Terraform to auto-generate non-overlapping subnets***

Need to attach an internet gateway and connect to the public subnet

A screenshot of a computer

AI-generated content may be incorrect.

Need to create or attach a subnet route table of public subnet route table. By default, nothing is created, so we shall add

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Make sure that the internet gateway is attached to VPC

A close-up of a computer screen

AI-generated content may be incorrect.

**Step 2: Create frontend for Web application**

Create the menu page, index page, Style and Frontend logic using Visual Studio code



Create an S3 Bucket

A screenshot of a computer

AI-generated content may be incorrect.

The bucket name should be unique

A screenshot of a computer

AI-generated content may be incorrect.

Block public access should be unchecked because we are gonna access this via public keep the rest of the settings enabled and create it.

A screenshot of a computer

AI-generated content may be incorrect.

Enable Static Website Hosting

Open your new bucket → Go to **Properties**

A screenshot of a computer

AI-generated content may be incorrect.

Scroll to Static website hosting and click Edit.

Enable it and set: **Index document** (index.html)

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Make the File Public**

Go to the Objects tab

Select index.html

Click Actions→ Make public

Confirm

**Access Your Website**

Go back to Static website hosting section in Properties

Copy the Bucket website endpoint

Paste it in your browser — your homepage should load.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Challenge Faced**

**Problem: Couldn’t access the website — received an “Access Denied” error.**

**Fix:**

**• Added a bucket policy to allow public access**

**• Enabled static website hosting**

**• Verified permissions and endpoint URL**

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* Go to your S3 bucket
* Click **Upload**
* Select all files from your Visual Studio project folder
* Click **Upload**
* After upload, go to **Permissions → Bucket Policy** (already set)
* Confirm all files are accessible via the **S3 website endpoint**

A screenshot of a computer

AI-generated content may be incorrect.

* Go to the **AWS Console → S3**
* Click on your bucket (e.g., al-den-cafe-frontend)
* Go to the **Properties** tab
* Scroll down to **Static website hosting**
* Look for the **“Bucket website endpoint”**

A screenshot of a computer

AI-generated content may be incorrect.

**Challenge Faced**

**Problem: Got a 404 error after opening the menu page.**

**Fix**

**• Identified mismatch in file reference — (lowercase) vs (capitalized)**

**• Corrected the link in the index file**

**• Re-uploaded the updated files to S3**

A close-up of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a menu

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Step 3: Creating the Backend**

We need to create a backend architecture for managing orders, user details etc. For which we are going for a Linux machine

Go to **AWS Console → EC2 → Launch Instance**

A screenshot of a computer

AI-generated content may be incorrect.

Name it: al-den-backend

Choose AMI: **Amazon Linux 2023**

A screenshot of a computer

AI-generated content may be incorrect.

Instance type: **t2.micro** (Free Tier)

Key pair:

Create new key pair

Download and save it securely

A screenshot of a computer

AI-generated content may be incorrect.

* Network settings:
* Select default VPC
* Allow inbound ports: **22** (SSH)
* Storage: Default 8 GB is fine
* Launch instance

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Step-by-Step: Connect to EC2 via SSH**

* Get Your Public IP
* Go to **EC2 Dashboard → Instances**
* Copy the **IPv4 Public IP**
* Open Windows Terminal (or PowerShell)
* Navigate to the folder where your .pem key is saved, then run:

ssh -i "pemfilename" ec2-user@public ip

**Challenge Faced**

**Problem: SSH access failed — the file couldn’t load due to permission issues.**

**Fix**

**• Terminated the Linux EC2 instance**

**• Pivoted to AWS Lambda for backend**

**• Simplified architecture by removing EC2 setup and SSH dependency**

**Building API with Lambda**

* Go to AWS Console → Lambda → Create Function
* Name: aldenMenuAPI
* Runtime **Python 3.12**
* Choose: **Author from scratch**
* Permissions: Create new role with basic Lambda permissions

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* Paste Your Code

def lambda\_handler(event, context):

menu\_items = [

{"name": "Coffee Latte", "price": "₹120"},

{"name": "Espresso", "price": "₹100"},

{"name": "Green Tea", "price": "₹90"},

{"name": "Masala Chai", "price": "₹80"},

{"name": "Cold Brew", "price": "₹130"}

]

return {

"statusCode": 200,

"headers": {"Content-Type": "application/json"},

"body": json.dumps(menu\_items)

}

A screenshot of a computer

AI-generated content may be incorrect.

3. Connect to API Gateway

Go to **API Gateway → Create API**

A screenshot of a computer

AI-generated content may be incorrect.

Choose **HTTP API**

A screenshot of a computer

AI-generated content may be incorrect.

* Add integration → select your Lambda function
* Add route: GET /menu
* Deploy and copy the public URL

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.