TECH GADGETS E-SHOP

A MICRO PROJECT REPORT

Submitted by

MOHAMUDHA SHIFA THAHSINA J

(Reg. No: 24MCR066)

SATHIYA JOTHI C

(Reg. No: 24MCR094)

SREENITHI G

(Reg. No: 24MCR106)

SUGUMAR M

(Reg. No: 24MCR109)

UMAMAHESHWARI M

(Reg. No: 24MCR120)

in partial fulfilment of the requirements for the

award of the degree

of

MASTER OF COMPUTER APPLICATIONS

DEPARTMENT OF COMPUTER APPLICATIONS

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI, ERODE – 638 060

MAY 2025

DEPARTMENT OF COMPUTER APPLICATIONS

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI ERODE-638 060

May 2025

BONAFIDE CERTIFICATE

Name	: **MOHAMUDHA SHIFA THAHSINA J(24MCR066)**
	SATHIYA JOTHI C(24MCR094)
	SREENITHI G(24MCR106)
	SUGUMAR M(24MCR109)
	UMAMAHESHWARI M(24MCR120)
Course Code	: **24MCT22**
Course Name	: **CLOUD COMPUTING TECHNOLOGY**
Semester	**II**

Certified that this is a bonafide record of work for application project done by the above students for **24MCT22- CLOUD COMPUTING TECHNOLOGY** during the academic year **2024-2025.**

Course Incharge **HEAD OF THE DEPARTMENT**

(Signature with seal)

Date:

Submitted for the viva-voce examination held on _____________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The "TechTrend Innovations" project was conceived as a practical exercise to solidify understanding and application of fundamental web development principles alongside cloud-based static website hosting. This initiative aimed to create a basic yet functional e-shop interface, with the long-term vision of displaying and potentially selling various tech gadgets. The initial phase, documented herein, focuses specifically on establishing the deployment infrastructure on Amazon Web Services (AWS) S3 and implementing robust local version control using Git.

1.1 OBJECTIVE

The core technical objectives of this project were twofold, addressing both the deployment process and foundational development practices. Firstly, the project aimed for the successful static website deployment on AWS S3, which involved several key steps. This included provisioning and configuring an S3 bucket specifically for hosting static web content, defining appropriate bucket policies to grant public read access to the necessary website assets, uploading the HTML and CSS files. Finally, verifying the accessibility of the deployed website through its S3-provided URL. Secondly, a parallel objective was to establish a robust local development environment that leverages Git for version control.

1.2 SCOPE

The scope of this initial phase of the "TechTrend Innovations" project was intentionally limited to the creation and deployment of a foundational static e-shop webpage. The specific focus included developing the basic structure and styling of the webpage utilizing HTML and CSS, configuring an AWS S3 bucket within the us-east-1 region for static website hosting as evidenced by the provided screenshots, uploading the core HTML file (index.html) to the designated S3 bucket, testing the public accessibility of the deployed webpage via its specific S3 URL, and initializing a local Git repository to effectively manage the project's source code.

It is crucial to note that the actual implementation of dynamic features such as shopping carts, payment integration, user authentication, inventory management, and any integration with backend services were considered outside the scope of this initial deployment phase, as reflected in the provided visual documentation which primarily showcases the successful configuration and deployment of the static hosting infrastructure on AWS S3.

1.3 TECHNOLOGIES USED: IN-DEPTH ANALYSIS

A more detailed examination of the technologies employed highlights their specific roles and significance within the project. HyperText Markup Language (HTML) served as the fundamental language for structuring the content displayed on the e-shop webpage, providing the semantic framework for product listings, navigation, and other sections. Cascading Style Sheets (CSS) were utilized to define the presentation and visual formatting of the HTML elements, controlling aspects such as layout, colors, fonts, and responsive design for various screen sizes. Amazon Web Services (AWS) Simple Storage Service (S3) provided the scalable and reliable infrastructure necessary for hosting the static website, offering high availability, durability, cost-effectiveness for static content, seamless integration with other AWS services, and global accessibility. Git was employed as the distributed version control system for managing the project's source code locally, enabling tracking of changes, collaboration, and efficient rollback capabilities.

SUMMARY

The "TechTrend Innovations" project was initiated as a practical exercise focused on deploying a static e-shop website to AWS S3 and implementing Git for version control. The primary technical objectives included successfully configuring an S3 bucket for static hosting, ensuring public accessibility of website assets, and establishing a local Git repository with a standard workflow. The core technologies employed were HTML for structure, CSS for styling, AWS S3 for static hosting, and Git for local code management. This foundational work sets the stage for future expansion and development of the "TechTrend Innovations" e-shop.

CHAPTER 2

SYSTEM REQUIREMENTS

This section outlines the software and tools required for the development and deployment of the "TechTrend Innovations" project, based on the technologies utilized and the processes observed in the provided screenshots.

2.1 SOFTWARE REQUIREMENTS

Based on the project's scope and the technologies employed, the following software would have been necessary for development:

- **Web Browser:** A modern web browser (such as Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge) was essential for viewing and testing the developed HTML and CSS webpage locally and for accessing the deployed website on AWS S3. The screenshots showing the AWS Management Console and the rendered webpage ("TECHTREND INNOVATIONS") indicate the use of a web browser.
- **Text Editor or Integrated Development Environment (IDE):** A text editor (like VS Code, Sublime Text, Atom, Notepad++) or an IDE (like IntelliJ IDEA or WebStorm) would have been used to write and edit the HTML and CSS code for the e-shop webpage. While the screenshots don't explicitly show the code editor, it is a fundamental requirement for web development.
- **Git:** The Git command-line interface (CLI) or a Git GUI client would have been required for initializing the local repository, committing changes, and potentially interacting with a remote repository. The inclusion of Git in the "Technologies Used" section and the Table of Contents confirms its necessity.

2.2 TOOLS AND PLATFORMS

The following tools and platforms were central to the deployment and infrastructure of the project, as evidenced by the screenshots:

- **AWS Management Console:** Access to the AWS Management Console via a web browser was essential for interacting with AWS services, specifically S3. The screenshots clearly show the user navigating the AWS console, creating an S3 bucket, configuring its properties, and uploading files.
- **Amazon Web Services (AWS) Simple Storage Service (S3):** As the core hosting platform for the static website, access to the S3 service within the AWS account was a fundamental requirement. The screenshots detail the creation and configuration of an S3 bucket named "techtrend" (and later "TechTrend innovations") in the us-east-1 (N. Virginia) region.
- **Git:** As mentioned in the software requirements, Git served as the local version control tool. While the platform for a potential remote repository (like GitHub, GitLab, or AWS CodeCommit) isn't explicitly shown, the use of Git implies the need for a Git hosting platform for collaboration and backup in a real-world scenario.

SUMMARY

The "TechTrend Innovations" project necessitated a standard web development software suite alongside the Amazon Web Services (AWS) platform for deployment and Git for version control. For development, a modern web browser was required for testing and accessing AWS services, a text editor or IDE for writing HTML and CSS, and Git for local version control. The core tools and platforms included the AWS Management Console for interacting with AWS services, specifically Amazon S3 for static website hosting in the us-east-1 region, and Git itself for managing the project's codebase locally. While not explicitly shown, a remote Git repository hosting platform would likely be used for collaboration and backup in a more comprehensive project setup.

CHAPTER 3

WEBPAGE DESIGN

This section details the design and structure of the "TechTrend Innovations" e-shop webpage, focusing on the underlying HTML structure and the styling applied using CSS. As the initial phase of the project involved deploying a foundational static webpage, the complexity of the design and styling might be basic, serving as a framework for future enhancements.

3.1 HTML STRUCTURE

This subsection will describe the organization of the content within the index.html file. It will outline the key HTML elements used to structure the webpage, including:

Document Structure: The fundamental <html>, <head>, and <body> tags that define the basic structure of the HTML document.

Metadata: Information within the <head> section, such as the <title> of the page, character encoding (<meta charset>), and crucial links to CSS stylesheets (<link>).

Content Organization: The use of semantic HTML5 elements (e.g., <header>, <nav>, <main>, <section>, <footer>) to structure the different sections of the e-shop webpage, such as the header with the site logo and navigation, product display sections, and a footer.

Content Elements: The specific HTML tags used to display content, such as headings (<h1> to <h6>) for product categories and names, paragraphs (<p>) for product descriptions, lists (<ul>, <ol>, <li>) for navigation or product specifications, images (<img>) for product visuals, and links (<a>) for navigation or "Add to Cart" actions.

Divs and Spans: The potential use of generic container elements (<div> and <span>) for structural or styling purposes, particularly for grouping product cards or other layout components.

Class and ID Attributes: The use of class and id attributes to apply CSS styles and potentially for JavaScript interaction in future iterations (e.g., .product-card, #main-navigation).

3.2 CSS STYLING

This subsection will describe how Cascading Style Sheets (CSS) were used to control the visual presentation of the HTML elements, making the "TechTrend Innovations" e-shop visually appealing and user-friendly. This will cover:

Stylesheet Inclusion: How the CSS rules were applied to the HTML, likely through an external stylesheet linked in the <head> section for maintainability and separation of concerns.

Selectors: The CSS selectors used to target specific HTML elements for styling (e.g., tag selectors, class selectors, ID selectors, descendant selectors).

Basic Styling Properties: The fundamental CSS properties likely used, such as:

Typography: font-family, font-size, color, text-align for titles, product names, prices, and descriptions.

Layout: display (e.g., block, inline-block, flex, grid) for arranging navigation items, product cards, and overall page structure. Properties like width, height, margin, padding for spacing and sizing.

Box Model: Understanding how margin, padding, border, and content contribute to the size and spacing of elements, crucial for well-defined product cards.

Backgrounds: background-color, background-image for the header, body, and distinct sections of the e-shop.

Borders and Shadows: For creating visually appealing product cards and buttons.

Transitions and Transforms: (Potentially for future enhancements) for subtle animations on hover effects on product cards or buttons.

Initial Visual Appearance: Based on the "TECHTREND INNOVATIONS" screenshot (Image 8), we can infer some basic styling such as the header background color, text styles for headings and product names, basic button styling, and the grid-like layout for product displays. This subsection will describe the likely CSS rules that would produce such an appearance.

Consideration for Responsive Design: The project description mentions "fully responsive." This indicates the use of media queries to adjust layout and styling for different screen sizes (desktops, tablets, mobile phones), ensuring an optimal viewing experience across various devices. This would involve adapting flex-direction, grid-template-columns, font sizes, and image scaling based on viewport width.

CODING

```
<!-- TechTrend Innovations - E-commerce Website -->

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>TechTrend Innovations</title>

</head>
```

```
<body style="margin: 0; font-family: 'Segoe UI', Arial, sans-serif; background-color: #f4f6f9;">

    <!-- Header Section -->

    <header style="background: linear-gradient(90deg, #1e3c72, #2a5298); padding: 15px 20px; color: white; box-shadow: 0 4px 6px rgba(0,0,0,0.1); position: sticky; top: 0; z-index: 100;">

        <div style="max-width: 1400px; margin: 0 auto; display: flex; justify-content: space-between; align-items: center;">

            <h1 style="margin: 0; font-size: 28px; text-transform: uppercase; letter-spacing: 1px;">TechTrend Innovations</h1>

            <nav style="display: flex; gap: 20px;">

                <a href="#login" style="color: white; text-decoration: none; font-size: 16px; transition: color 0.3s ease;">Login</a>

                <a href="#cart" style="color: white; text-decoration: none; font-size: 16px; transition: color 0.3s ease;" onclick="showCart()">Cart (<span id="cart-count">0</span>)</a>

            </nav>

        </div>

    </header>



    <!-- Main Content -->

    <main style="max-width: 1400px; margin: 0 auto; padding: 40px 20px;">

        <!-- Home Section -->
        <section id="home">
          <div style="display: flex; justify-content: space-between; align-items: center; margin-bottom: 30px;">
              <h2 style="font-size: 28px; color: #333;">Featured Products</h2>
              <div>
                  <select id="category-filter" style="padding: 8px; border-radius: 5px; border: 1px solid #ccc; font-size: 14px;">
                      <option value="all">All Categories</option>
                      <option value="smartphone">Smartphones</option>
                      <option value="tablet">Tablets</option>
                      <option value="laptop">Laptops</option>
                      <option value="accessory">Accessories</option>
                 </select>
              </div>
           </div>
           <div id="product-grid" style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 20px;">
              <!-- Product 1 -->
<div class="product" data-category="Laptops" style="background: white; border-radius: 10px; overflow: hidden; box-shadow: 0 4px 8px rgba(0,0,0,0.1); transition: transform 0.3s ease, box-shadow 0.3s ease;">
  </div>
```

```
</div>
            <!-- Product 2 -->
   <div style="padding: 15px; text-align: center;">
      <h3 style="margin: 0; font-size: 18px; color: #333;">Sony WF</h3>
      <p style="color: #777; margin: 5px 0;">$299.99</p>
             </div>
             <!-- Product 5 -->
                <div class="product" data-category="smartphone" style="background: white; border-radius: 10px; overflow: hidden; box-shadow: 0 4px 8px rgba(0,0,0,0.1); transition: transform 0.3s ease, box-shadow 0.3s ease;">
<img src="https://www.cnet.com/a/img/resize/bde1b8ca1b9373b61bbf9d3e113a81ac76297b51/hub/2024/09/13/0df30744-a33f-4c6e-b58c-a90d7a914089/apple-iphone-16-2815.jpg?auto=webp&height=500" alt="iPhone" style="width: 100%; height: 200px; object-fit: cover;">
                <div style="padding: 15px; text-align: center;">
                   <h3 style="margin: 0; font-size: 18px; color: #333;">iPhone 16</h3>
                   <p style="color: #777; margin: 5px 0;">$899.99</p>
                </div>
             </div>
             <!-- Product 6 -->
                   <h3 style="margin: 0; font-size: 18px; color: #333;">AirPods Pro</h3>
```

```
          <p style="color: #777; margin: 5px 0;">$249.99</p>
            <button onclick="addToCart('AirPods Pro', 249.99)" style="background: #2a5298; color: white; border: none; padding: 10px 20px; border-radius: 5px; cursor: pointer; transition: background 0.3s ease;">Add to Cart</button>
      </div>
    </div>
    </div>
    <!-- Product 8 -->
      <div style="padding: 15px; text-align: center;">
        <h3 style="margin: 0; font-size: 18px; color: #333;">OnePlus Nord CE</h3>
      </div>
    </div>
    <!-- Product 9 -->
     <div class="product" data-category="tablet" style="background: white; border-radius: 10px; overflow: hidden; box-shadow: 0 4px 8px rgba(0,0,0,0.1); transition: transform 0.3s ease, box-shadow 0.3s ease;">
<img src="https://5.imimg.com/data5/SELLER/Default/2023/12/372325508/ZZ/TZ/XX/180833002/tab-m10-25-65cms-10-1-4gb-64gb-storm-grey-500x500.png" alt="Tablet" style="width: 100%; height: 200px; object-fit: cover;">
      </div>
    </div>
    <!-- Product 10 -->
```

```
        cart.push({ name, price });
        updateCart();
    }
    // Update Cart Display
    function updateCart() {
        const cartItems = document.getElementById('cart-items');
        const cartCount = document.getElementById('cart-count');
        const cartTotal = document.getElementById('cart-total');
        cartCount.textContent = cart.length;
        if (cart.length === 0) {
            cartItems.innerHTML = '<p style="text-align: center; color: #777;">Your cart is empty.</p>';
            cartTotal.textContent = '0.00';
            return;
        }
        cartItems.innerHTML = cart.map((item, index) => `
            <div style="display: flex; justify-content: space-between; padding: 10px; border-bottom: 1px solid #eee;">
                <span>${item.name}</span>
                <span>$${item.price.toFixed(2)}</span>
                <button onclick="removeFromCart(${index})" style="background: #d32f2f; color: white; border: none; padding: 5px 10px; border-radius: 5px; cursor: pointer;">Remove</button>
            </div>
        `).join('');
        const total = cart.reduce((sum, item) => sum + item.price, 0);
        cartTotal.textContent = total.toFixed(2);
    }
    // Remove from Cart
    function removeFromCart(index) {
        cart.splice(index, 1);
        updateCart();
    }
    // Show Cart
    function showCart() {
        document.getElementById('cart').style.display = 'block';
        document.getElementById('home').style.display = 'none';
        document.getElementById('about').style.display = 'none';
        document.getElementById('login').style.display = 'none';
        window.scrollTo({ top: 0, behavior: 'smooth' });
    }
```

```
// Smooth Scroll for Navigation
document.querySelectorAll('nav a').forEach(anchor => {
  anchor.addEventListener('click', function(e) {
    if (this.getAttribute('href') !== '#cart') {
      e.preventDefault();
      const targetId = this.getAttribute('href').substring(1);
      const targetSection = document.getElementById(targetId);
      document.querySelectorAll('main > section').forEach(section => {
        section.style.display = section.id === targetId ? 'block' : 'none';
      });
      window.scrollTo({ top: 0, behavior: 'smooth' });
    }
  });
});
// Category Filter
document.getElementById('category-filter').addEventListener('change', function() {
  const category = this.value;
  const products = document.querySelectorAll('.product');
  products.forEach(product => {
```

```
            product.style.display = category === 'all' || product.dataset.category === category ? 'block' : 'none';

          });

        });

        // Initial Setup

        document.getElementById('home').style.display = 'block';

        document.getElementById('about').style.display = 'none';

        document.getElementById('login').style.display = 'none';

        document.getElementById('cart').style.display = 'none';

    </script>

</body>

</html>
```

SUMMARY

The initial "TechTrend Innovations" e-shop utilized a structured HTML foundation for product displays, navigation, and other content, combined with CSS for comprehensive styling. The deployed version, as seen in the provided screenshots, demonstrates a visually appealing layout with distinct product cards, a clear header, and a responsive design approach. The HTML likely included core semantic elements, while CSS was applied (likely externally) for typography, flexible and grid-based layouts, background appearances, and responsive adjustments via media queries. This robust front-end structure serves as the foundation for the e-shop, ready for future content expansion and potential interactive features.

CHAPTER 4

DEPLOYMENT ON AWS S3

This section outlines the step-by-step process of deploying the Country Profile Info Page on Amazon Web Services (AWS) using S3 (Simple Storage Service), which is a cost effective and reliable solution for hosting static websites.

4.1 CREATING AN S3 BUCKET

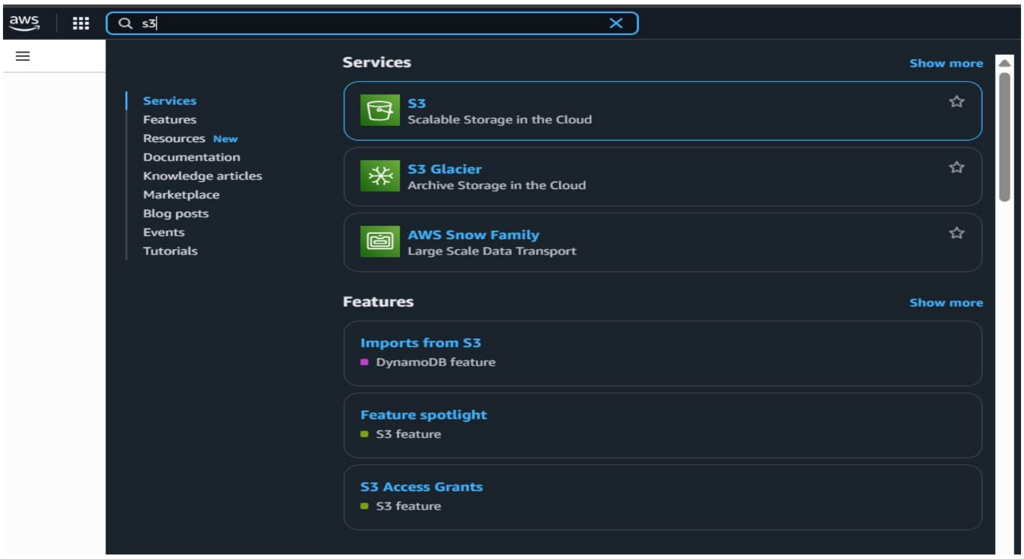
1. Log in to your AWS Management Console.

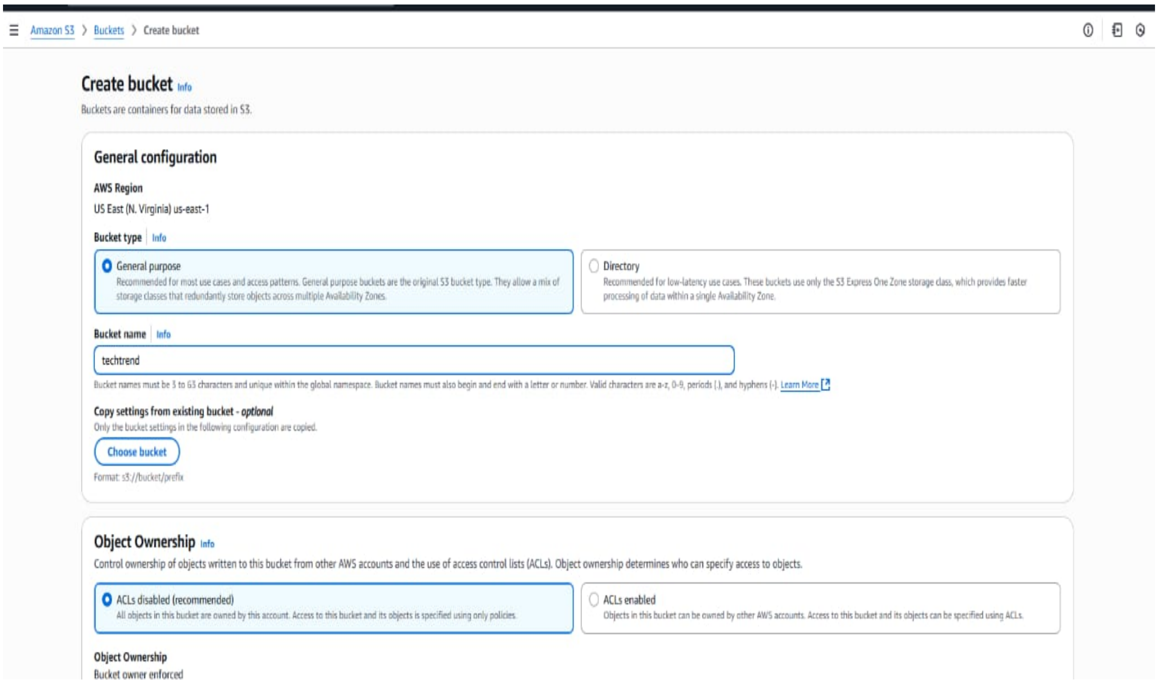
2. Navigate to the S3 service from the Services menu.

3. Click on the "Create bucket" button.

4. Enter a unique bucket name (e.g., country-profile-info-page) and select your preferred AWS Region.

5. Keep default settings for most options, but uncheck the box that blocks all public acces





Create bucket

Buckets are containers for data stored in S3.

General configuration

AWS Region

US East (N. Virginia) us-east-1

Bucket type

● General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

○ Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name

techtrend

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-).

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Format: s3://bucket/prefix

Object Ownership

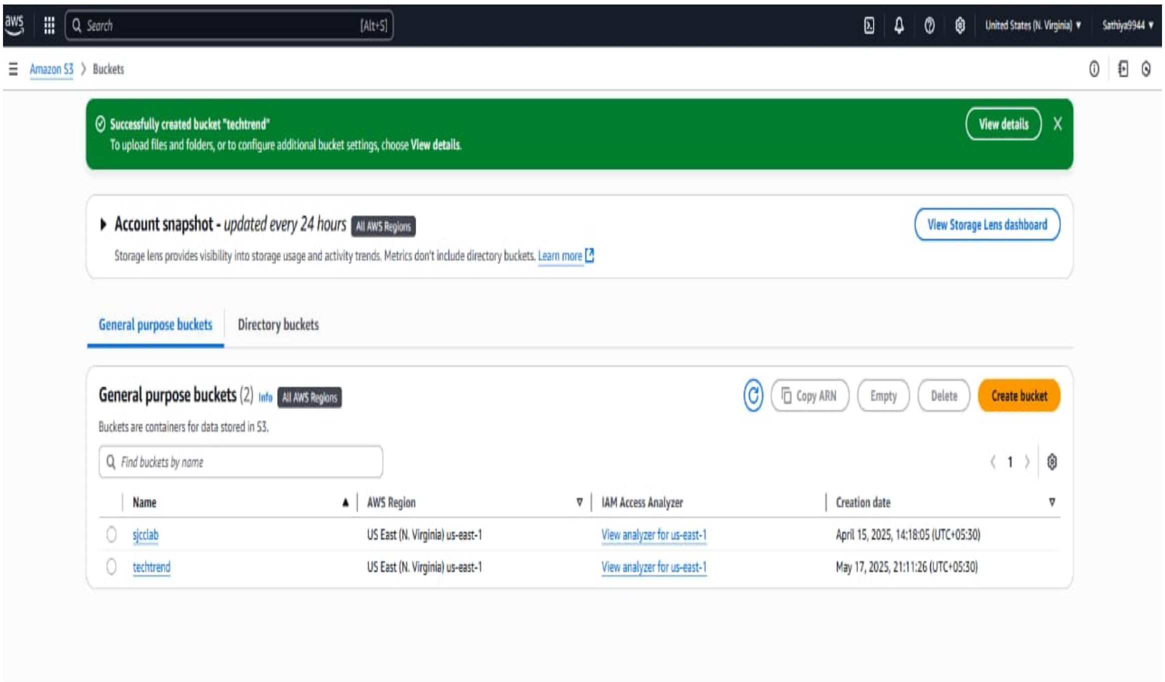
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

● ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

○ ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

Bucket owner enforced



Successfully created bucket "techtrend"

To upload files and folders, or to configure additional bucket settings, choose **View details**.

Account snapshot - *updated every 24 hours*

Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets.

General purpose buckets (2)

Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
○	sjcclab	US East (N. Virginia) us-east-1	View analyzer for us-east-1	April 15, 2025, 14:18:05 (UTC+05:30)
○	techtrend	US East (N. Virginia) us-east-1	View analyzer for us-east-1	May 17, 2025, 21:11:26 (UTC+05:30)

4.2 CONFIGURING BUCKET FOR STATIC WEBSITE HOSTING

1. Once the bucket is created, click on the bucket name to open it.

2. Go to the "Properties" tab.

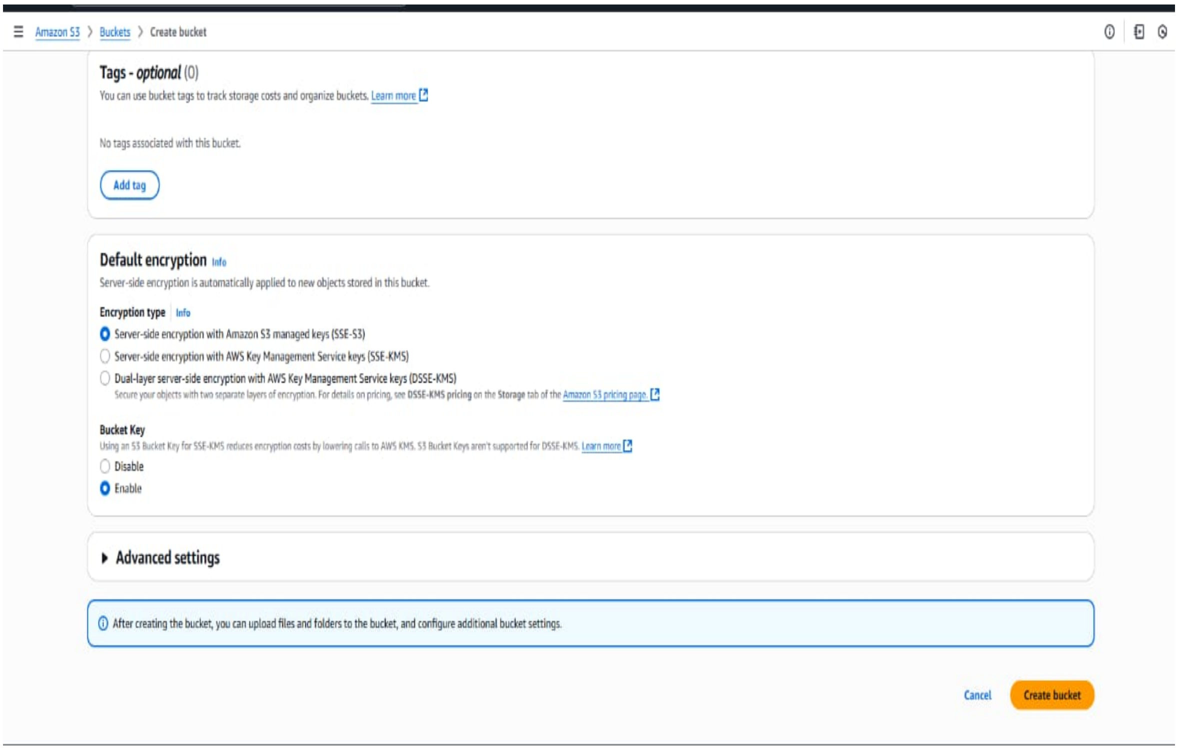
3. Scroll down to find the "Static website hosting" section.

4. Click "Edit" and enable static website hosting.

5. Choose “Host a static website”.

6. In the fields provided: o Index document: index.html o Error document (optional): error.html.

7. Save the changes.



Tags - *optional* (0)

You can use bucket tags to track storage costs and organize buckets. Learn more

No tags associated with this bucket.

Default encryption Info

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type | Info

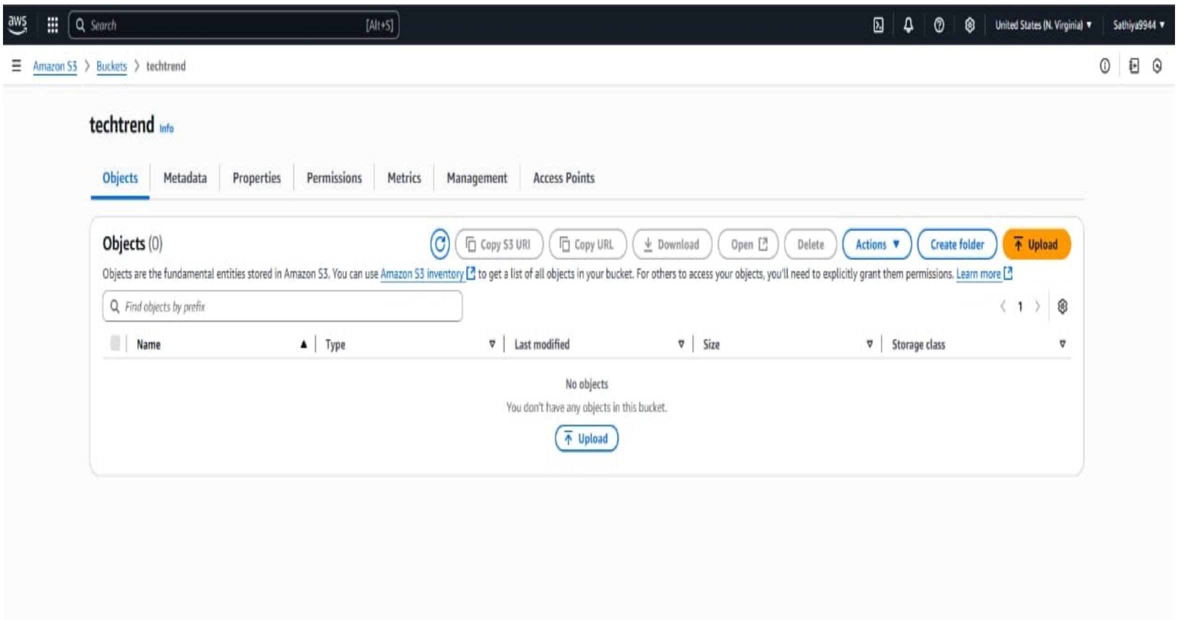
● Server-side encryption with Amazon S3 managed keys (SSE-S3)
○ Server-side encryption with AWS Key Management Service keys (SSE-KMS)
○ Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see **DSSE-KMS pricing** on the **Storage** tab of the Amazon S3 pricing page.

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more

○ Disable
● Enable

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

No objects

You don't have any objects in this bucket.

4.3 UPLOADING FILES TO S3

1. Go to the "Objects" tab of your S3 bucket.

2. Click "Upload" and then "Add files”.

3. Select your project files:

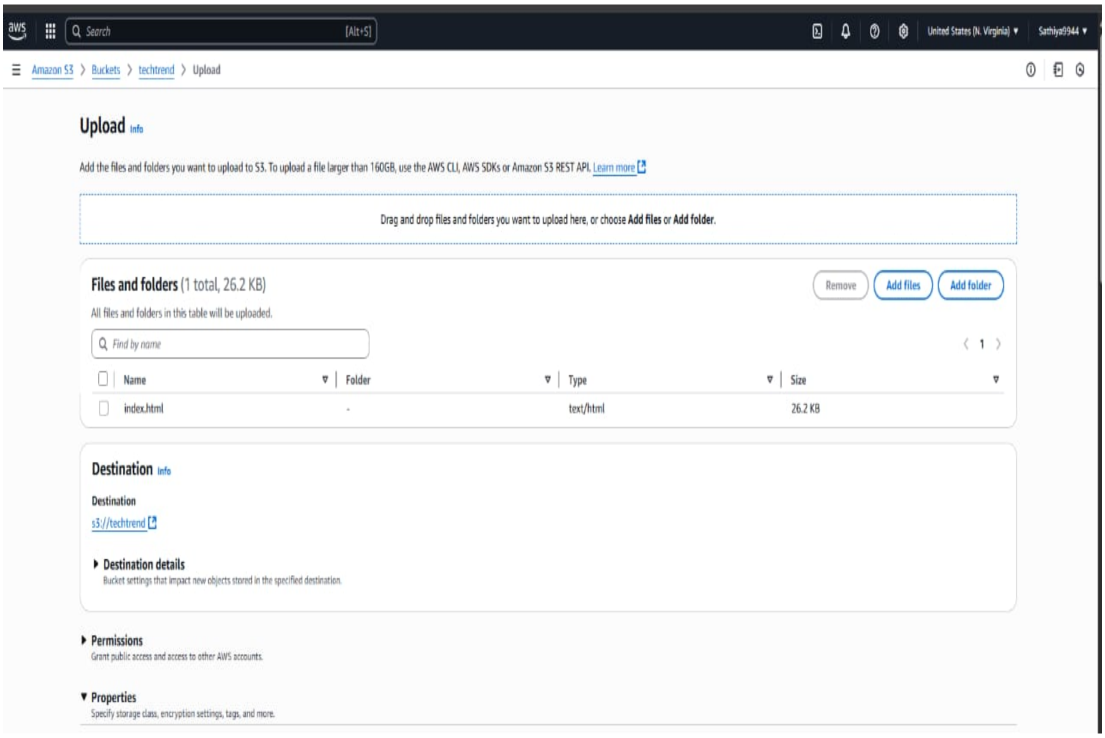
4. index.html

5. style.css

6. Any images or assets (e.g., flag, map image, etc.)

7. Click "Upload" to finish the process.

8. After upload, click on each file (especially index.html) and choose “Make public” (unless you’ve set bucket-level public access).



aws
Search [Alt+S]
United States (N. Virginia) ▾
Sathiya9944 ▾
Amazon S3 > Buckets > techtrend > Upload

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (1 total, 26.2 KB)

All files and folders in this table will be uploaded.

□	Name	Folder	Type	Size
□	index.html	-	text/html	26.2 KB

Destination Info

Destination

s3://techtrend

▸ **Destination details**

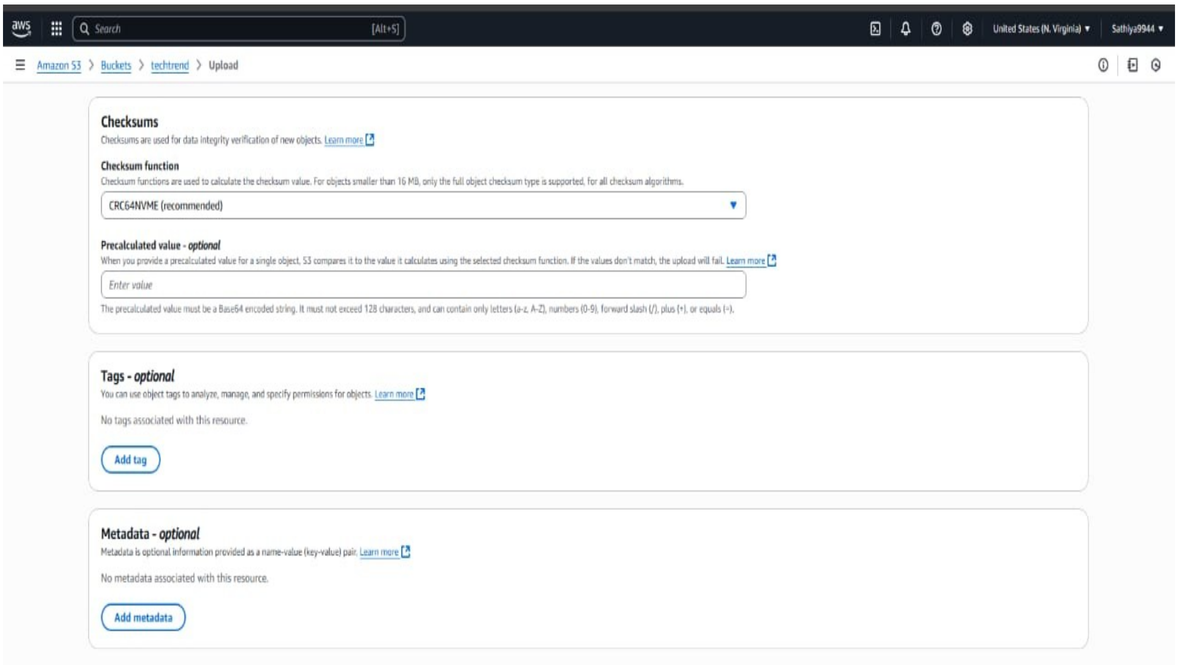
Bucket settings that impact new objects stored in the specified destination.

▸ **Permissions**

Grant public access and access to other AWS accounts.

▾ **Properties**

Specify storage class, encryption settings, tags, and more.



aws
Search [Alt+S]
United States (N. Virginia) ▾
Sathiya9944 ▾
Amazon S3 > Buckets > techtrend > Upload

Checksums

Checksums are used for data integrity verification of new objects. Learn more

Checksum function

Checksum functions are used to calculate the checksum value. For objects smaller than 16 MB, only the full object checksum type is supported, for all checksum algorithms.

CRC64NVME (recommended) ▾

Precalculated value - *optional*

When you provide a precalculated value for a single object, S3 compares it to the value it calculates using the selected checksum function. If the values don't match, the upload will fail. Learn more

The precalculated value must be a Base64 encoded string. It must not exceed 128 characters, and can contain only letters (a-z, A-Z), numbers (0-9), forward slash (/), plus (+), or equals (=).

Tags - *optional*

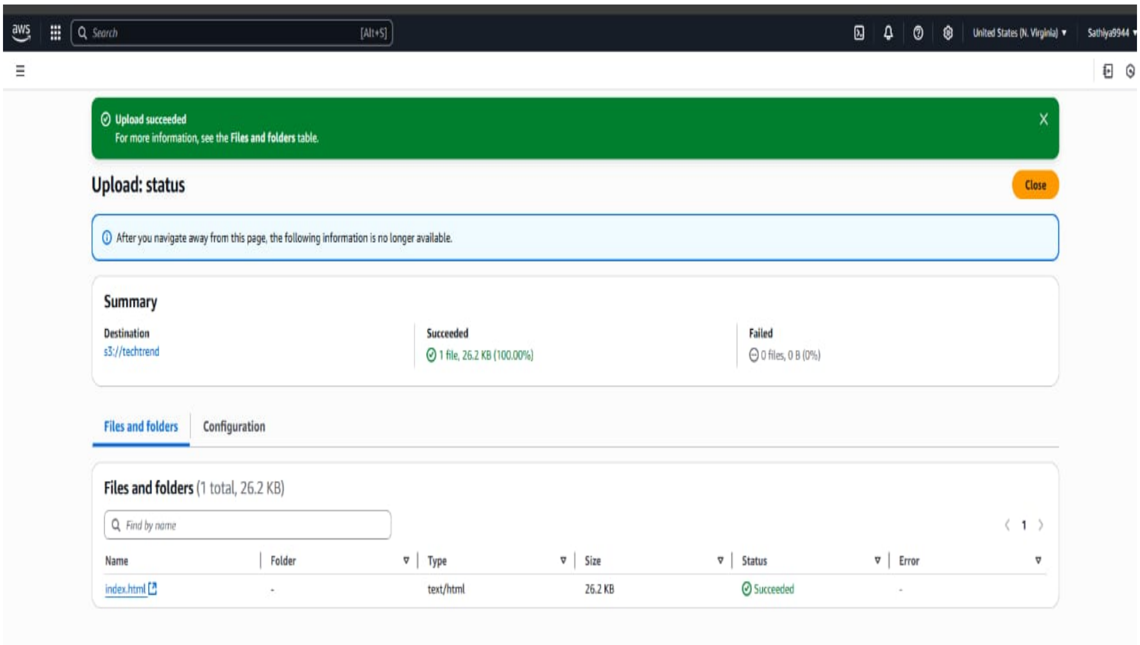
You can use object tags to analyze, manage, and specify permissions for objects. Learn more

No tags associated with this resource.

Metadata - *optional*

Metadata is optional information provided as a name-value (key-value) pair. Learn more

No metadata associated with this resource.



Upload succeeded
For more information, see the Files and folders table.

Upload: status

Close

After you navigate away from this page, the following information is no longer available.

Summary

Destination: s3://techtrend

Succeeded: 1 file, 26.2 KB (100.00%)

Failed: 0 files, 0 B (0%)

Files and folders | Configuration

Files and folders (1 total, 26.2 KB)

Name	Folder	Type	Size	Status	Error
index.html	-	text/html	26.2 KB	Succeeded	-

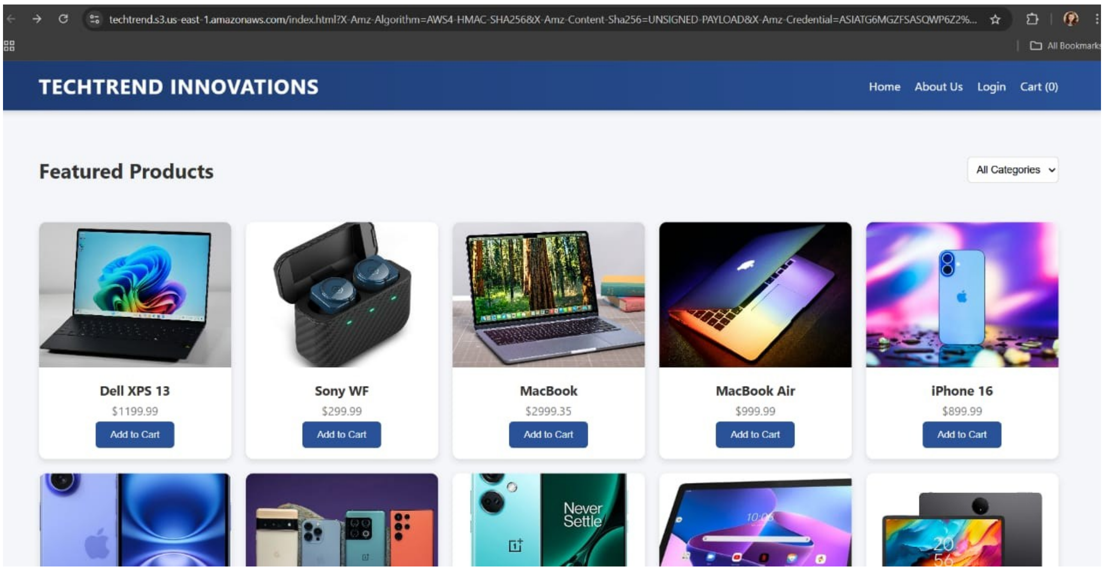
4.4 TESTING THE DEPLOYMENT

1. Return to the "Properties" tab of your bucket.

2. In the Static website hosting section, find the “Bucket website endpoint” (e.g., http://your-bucket-name.s3-website-region.amazonaws.com).

3. Click or copy this URL and open it in your browser.

4. Your Country Profile Info Page should now be live and publicly accessible.



techtrend.s3.us-east-1.amazonaws.com/index.html?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Content-Sha256=UNSIGNED-PAYLOAD&X-Amz-Credential=ASIATG6MGZFSASQWP6Z2%...

TECHTREND INNOVATIONS

Home About Us Login Cart (0)

Featured Products

All Categories

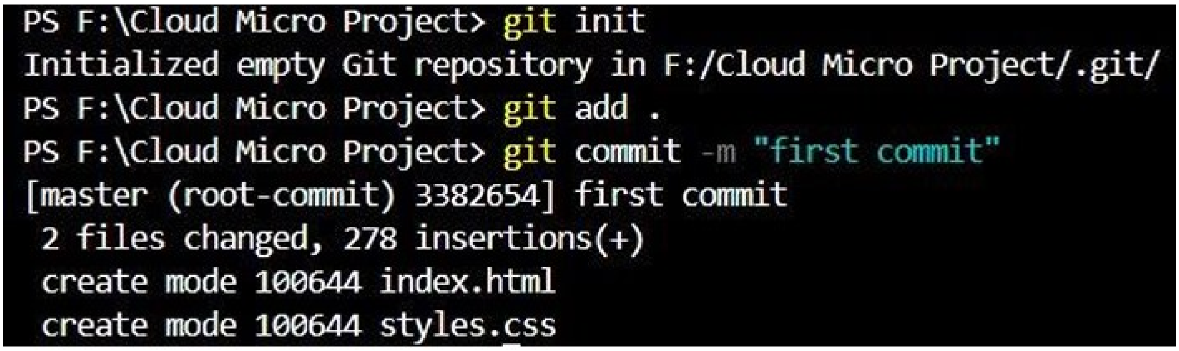
Dell XPS 13	Sony WF	MacBook	MacBook Air	iPhone 16
$1199.99	$299.99	$2999.35	$999.99	$899.99
Add to Cart	Add to Cart	Add to Cart	Add to Cart	Add to Cart

5. VERSION CONTROL WITH GIT

Version control is a crucial part of modern software development, allowing developers to manage changes, track progress, and collaborate efficiently. In this project, Git is used as the version control system, and GitHub is the remote hosting platform. This section explains the process of initializing the Git repository, committing and pushing code, and the final structure of the repository.

5.1 INITIALIZING GIT REPOSITORY

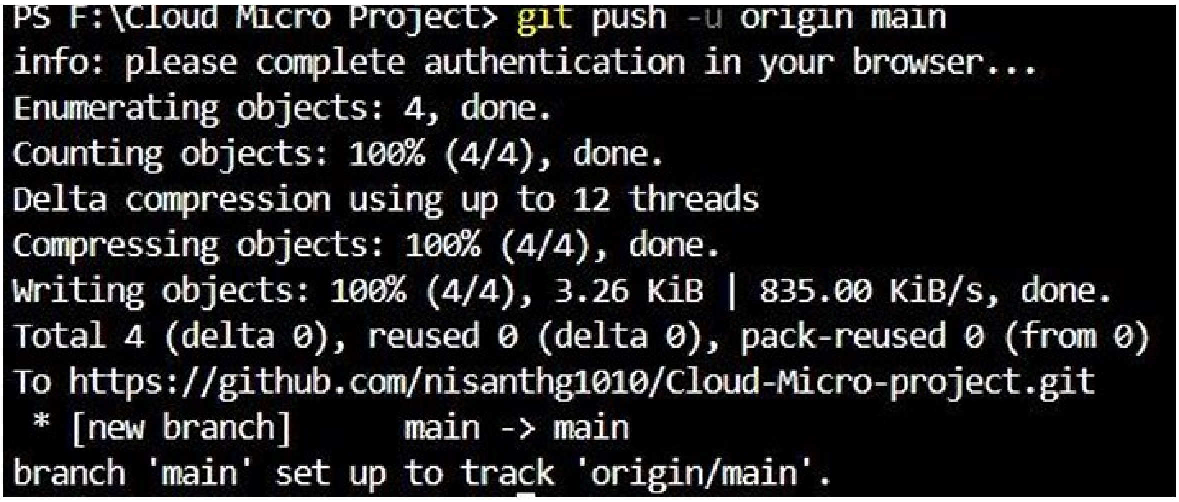
The first step in implementing version control with Git is initializing a Git repository in the local project directory. This process involves creating a hidden directory (.git) within the project folder that will store metadata and object information needed to track changes to the files. This enables the local environment to recognize and monitor all modifications made to the project's source code. Initialization ensures that the entire development process is documented and that changes can be reverted or reviewed at any stage of the project lifecycle.



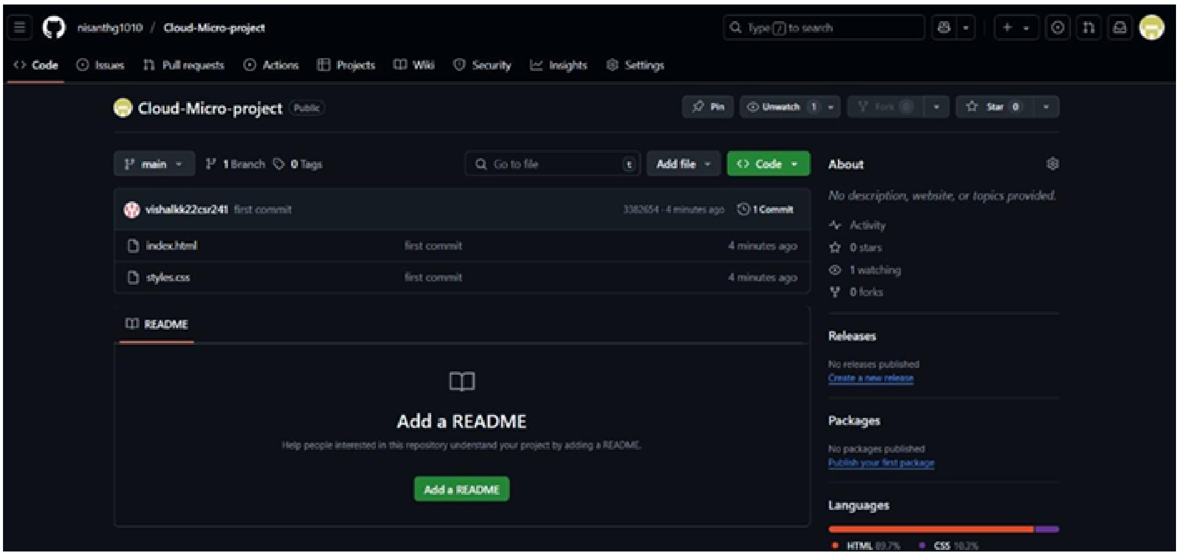
```
PS F:\Cloud Micro Project> git init
Initialized empty Git repository in F:/Cloud Micro Project/.git/
PS F:\Cloud Micro Project> git add .
PS F:\Cloud Micro Project> git commit -m "first commit"
[master (root-commit) 3382654] first commit
 2 files changed, 278 insertions(+)
 create mode 100644 index.html
 create mode 100644 styles.css
```

5.2 COMMITTING AND PUSHING CODE

After initializing the repository, the next step is staging and committing code changes. Staging selects specific files to save, while committing records them with a message. Pushing to GitHub backs up the code remotely and supports collaboration.



```
PS F:\Cloud Micro Project> git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 3.26 KiB | 835.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/nisanthg1010/Cloud-Micro-project.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```



vishalkk22csr241 first commit 3382654 · 4 minutes ago 1 Commit

index.html first commit 4 minutes ago

styles.css first commit 4 minutes ago

Help people interested in this repository understand your project by adding a README.

No description, website, or topics provided.

0 stars

1 watching

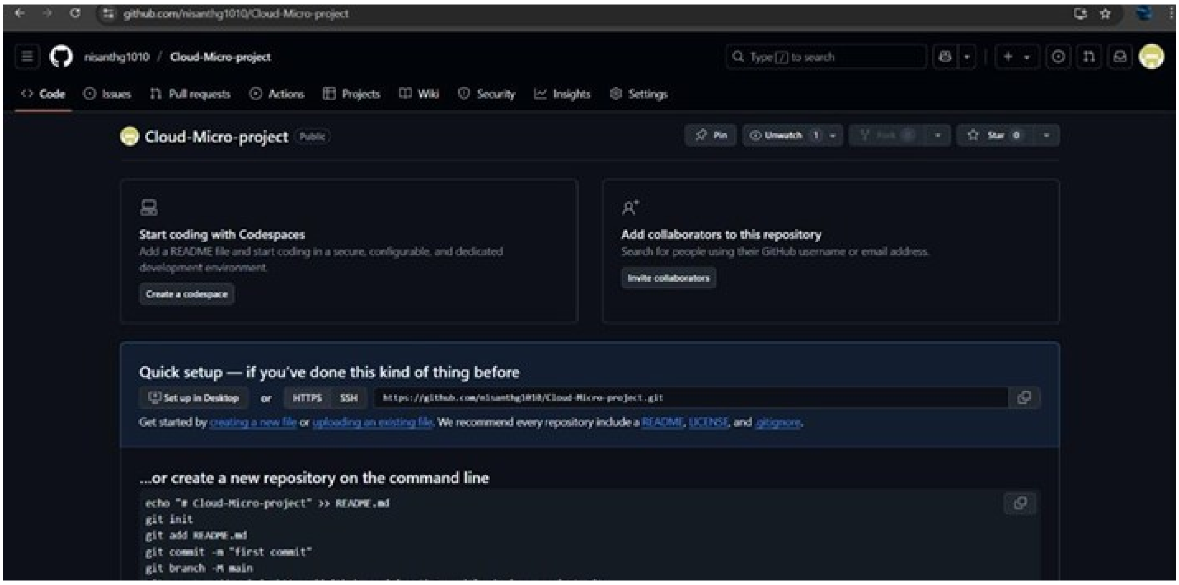
0 forks

No releases published
Create a new release

No packages published
Publish your first package

5.3 REPOSITORY STRUCTURE

Maintaining a clear and organized repository structure is vital for readability, maintainability, and scalability. In this project, the repository is structured to separate content (HTML), styling (CSS), and assets (such as images) into logical directories and files. A README.md file is included to offer documentation and project context, while a .gitignore file prevents unnecessary or sensitive files from being tracked. This structured approach allows developers and collaborators to easily navigate the project, understand its components, and contribute effectively. The hosted repository on GitHub further enables version history access, collaborative coding, and transparent project sharing.



Start coding with Codespaces

Add a README file and start coding in a secure, configurable, and dedicated development environment.

Add collaborators to this repository

Search for people using their GitHub username or email address.

Quick setup — if you've done this kind of thing before

https://github.com/nisanthg1010/Cloud-Micro-project.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# Cloud-Micro-project" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
```

6. CONCLUSION

The TechTrend – Gadgets E-Shop is a fully responsive, static e-commerce website developed to showcase and sell the latest tech gadgets in a clean, user-friendly format. Built using foundational front-end technologies—HTML for structure, CSS for styling, and optionally JavaScript for interactive elements—the platform focuses on delivering a smooth and intuitive shopping experience across all devices. Key features include categorized product displays, brief specifications, pricing details, and visually distinct product cards that allow users to browse and identify items of interest quickly and efficiently.

The layout leverages modern CSS techniques such as flexbox and grid to ensure content is well-organized and adjusts seamlessly to different screen sizes, enhancing accessibility for users on desktops, tablets, and mobile phones. The project is aimed at both tech enthusiasts and casual buyers, providing a fast and distraction-free way to explore trending gadgets.

For version control and collaboration, Git is used to track code changes, while GitHub serves as the remote repository, enabling efficient management of development history and contributions. Deployment is handled using AWS S3 static website hosting, offering high availability, global accessibility, and cost-effective performance. This choice demonstrates not only practical knowledge of cloud hosting but also familiarity with basic DevOps workflows such as versioning, updates, and environment management.

In essence, TechTrend represents a practical implementation of core web development concepts combined with industry-aligned deployment practices. The project serves as a solid foundation for more advanced features like shopping carts, payment integration, user authentication, and backend connectivity—making it an ideal starting point for aspiring developers and a scalable solution for digital storefronts in the tech retail space.

7. REFERENCES

[1] W3C, "HTML5 Specification," World Wide Web Consortium (W3C), 2014. [Online]. Available: https://www.w3.org/TR/html5/

[2] W3C, "Cascading Style Sheets (CSS) - The Official Definition," World Wide Web Consortium (W3C), 2023. [Online]. Available: https://www.w3.org/Style/CSS/

[3] Git SCM, "Git - Distributed Version Control System," Git, 2024. [Online]. Available: https://git-scm.com/

[4] GitHub Inc., "GitHub: Where the world builds software," GitHub, 2024. [Online]. Available: https://github.com/

[5] E. Freeman and E. Robson, Head First HTML and CSS, 2nd ed., O'Reilly Media, 2012.
[6] B. Beighley, Responsive Web Design with HTML5 and CSS, 2nd ed., Peachpit Press, 2015.

[7] Mozilla Developer Network (MDN), "HTML: HyperText Markup Language," MDN Web Docs, 2024. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTML

[8] M. Fowler, "Continuous Integration," martinfowler.com, 2006. [Online]. Available: https://martinfowler.com/articles/continuousIntegration.html

[9] Amazon Web Services, "Getting Started with AWS – Hosting a Static Website," AWS Getting Started Resource Center, 2023. [Online]. Available: https://aws.amazon.com/getting started/hands-on/host-static-website/

[10] J. Duckett, HTML and CSS: Design and Build Websites, 1st ed., Indianapolis, IN, USA: Wiley, 2011.