**Hosting E-Commerce Website in AWS**

**Project Synopsis**

Industrial Training (ECS591)

Degree

**BACHELOR OF TECHNOLOGY (CSE)**

|  |  |
| --- | --- |
| PROJECT GUIDE:  **Mohan V.Gupta (Internal)** | SUBMITTED BY:  **Harsh Madhur (TCA2211011)**  **Aman Kumar TCA2211002))** |

DEC, 2023



**FACULTY OF ENGINEERING & COMPUTING SCIENCES**

**TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD**

Table of Contents

[1 Project Title 3](#_Toc31139962)

[2 Problem Statement 3](#_Toc31139964)

[3 Project Description 3](#_Toc31139965)

[3.1 Scope of the Work 3](#_Toc31139966)

[3.2 Project Modules 3](#_Toc31139967)

[4 Implementation Methodology 3](#_Toc31139968)

[5 Technologies to be used 4](#_Toc31139969)

[5.1 Software Platform 4](#_Toc31139970)

[5.2 Hardware Platform 4](#_Toc31139971)

[6 Advantages of this Project 4](#_Toc31139973)

[7 Future Scope and further enhancement of the Project 4](#_Toc31139974)

[8 Team Details 4](#_Toc31139975)

[9 Conclusion 5](#_Toc31139976)

[10 References 5](#_Toc31139977)

# 1.Project Title

We are all aware that technology has evolved into an essential tool for online marketing in the modern world .We may infer that the majority of people around the world are enthusiastic about online shopping. However, its also clear that a sizable number of corner stores and grocery stores offer their goods offline. Your online business can be supported by a scalable, secure, and reliable infrastructure if you host your e-commerce In this project we build a E-commerce website for hosting, in this website we’re showing some IT Product, all this looks very nicely. A static website delivers content in the same format in which it is stored .No server-side code execution is required. For example, if a static website consists of HTML documents displaying images, it delivers the HTML and imagb server doesn’t need to perform any application logic or database ques as-is to the browser, without altering the contents of the files. Static websites can be delivered to web browsers on desktops, tablets, or mobile devices. They usually consist of a mix of HTML documents ,images, videos, CSS style sheets, and JavaScript files.AWS provides reliable, scalable, secure and highly performing infrastructure for the most demanding web application. This infrastructure matches IT cost with customer traffic pattern in real time. The traditional web hosting architecture that implements a common 3 tier web application model. Static hosting provides the users, only the previously recorded data, whereas for any restaurants or for any ecommerce websites dynamic hosting should be required for the users to get to know what about the daily/weekly offers and all. By using AWS we can provide instance i.e. website instant link for the users who can't be able to access the website due to any reason. The main purpose of this project is to develop a E-commerce (harsh Website) and to deploy the website into the cloud using AWS like EC2, S3, IIS and VPC we establish the instance and send it to the required users and also generate the reports regarding the strength of the website to the users. There are 3 main processes in this project. Which are designing a website and uploading/deploying the website into IIS and connecting website instances with EC2.

# 2.Problem Statement

Slow connections: Check the instance's system log for errors like exhausted memory or a full disk. Also, make sure the web service is running and started correctly. Connection issues: EC2 Instance Connect might not work as expected if: The OS distribution isn't supported The instance doesn't have the EC2 Instance Connect package installed There are missing or incorrect AWS Identity and Access Management (IAM) policies or permissions Internet gateway connection issues: Make sure the EC2 instance meets all prerequisites, has a public IP address, and that a firewall isn't blocking access Instance launch failures: Some options in the launch template or launch configuration might not be compatible with the instance type. The instance configuration might also not be supported in the requested AWS Region or Availability Zones. Hardware-level changes: These can result in poor performance and usage of applications Security constraints: AWS restricts some of its features that cannot be changed at all

# 3.Project Description

1.AWS Account Setup: Create an AWS account if you don't have one. Set up IAM (Identity and Access Management) users with appropriate permissions. Generate and download access keys for programmatic access.

2.Launch an EC2 Instance : Navigate to the EC2 dashboard. Launch a new EC2 instance, choosing an Amazon Machine Image (AMI) that suits your e-commerce platform requirements (e.g., Amazon Linux, Ubuntu).Configure instance details, add storage, and configure security groups (open ports like 80 for HTTP, 443 for HTTPS). Launch the instance and associate it with a key pair for SSH access.

3.Connect to the EC2 Instance: Use an SSH client to connect to your EC2 instance using the key pair. Update the system and install necessary software (e.g., web server, database, programming language runtime).

4.Install Web Server and Configure: Install a web server like Apache or Nginx. Configure the server to serve your e-commerce application. Set up virtual hosts if hosting multiple sites.

5.Install and Configure Database: Choose a suitable database (e.g., MySQL, PostgreSQL). Install and configure the database server. Create a database and user for your e-commerce application.

6.Deploy E-Commerce Application: Transfer your e-commerce application code to the EC2 instance. Configure application settings, database connections, and any necessary environment variables.

7.Secure the EC2 Instance: Configure the firewall rules (security groups) to restrict access. Set up SSL/TLS certificates for secure HTTPS connections. Consider using AWS Identity and Access Management (IAM) for fine-grained access control.

8.Backup and Monitoring: Implement regular backups of your database and application data. Set up monitoring tools (e.g., AWS CloudWatch) to track performance and receive alerts.

9.Scale and Load Balancing : Consider implementing auto-scaling and load balancing for high availability and improved performance.

10.Domain Name and DNS Setup: Register a domain name if you haven't already. Configure DNS settings to point to your EC2 instance's public IP or Elastic Load Balancer.

11.Testing:Test your e-commerce website thoroughly to ensure functionality and performance. Consider implementing a staging environment for pre-production testing.

12.Continuous Integration and Deployment (Optional):Set up a CI/CD pipeline for automating code deployment.

13.Documentation and Training : Document the configuration and setup for future reference. Provide training materials for maintenance and troubleshooting.

## Scope of the Work

* *Buying and selling*
* *Serving customers*
* *Processing business payments*
* *Managing production control*
* *Collaborating with business partners*
* *Sharing information via the internet*
* *Running automated employee services*
* *Recruiting*

## 3.2Project Modules

1.User Authentication and Authorization:

* User registration
* Login and logout
* Password recovery
* User roles and permissions

2.Product Management:

* Add, edit, and delete products
* Product categories and attributes
* Product search and filtering
* Inventory management

3.Shopping Cart:

* Add products to the cart
* Update or remove items from the cart
* View and manage the shopping cart
* Save the cart for future sessions

4.Order Processing:

* Checkout process
* Billing and shipping information
* Order confirmation and summary
* Payment gateway integration

5.Payment Gateway:

* Integration with payment services (e.g., PayPal, UPI)
* Handling payment transactions securely

6.User Profile:

* User dashboard
* Order history and tracking
* Wishlist and favorites
* Reviews and Ratings:
* Allow users to leave product reviews
* Implement a rating system

7.Search and Filters:

* Implement a robust search functionality
* Filtering products based on various criteria

8.Security:

* Secure data transmission (SSL)
* Data encryption and protection
* Protection against common web vulnerabilities

9.Admin Panel:

* Dashboard with key metrics
* User management
* Order management
* Content management (for static pages, banners, etc.)

10.Email Notifications:

* Order confirmation emails
* Shipping notifications
* Marketing and promotional email

# 4.Implementation Methodology

Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts:

* Model - The lowest level of the pattern which is responsible for maintaining data.
* View - This is responsible for displaying all or a portion of the data to the user.
* Controller - Software Code that controls the interactions between the Model and View.

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows.

E-R Diagram: - An Entity Relationship Diagram is a diagram that represents relationships among entities in a database An entity–relationship model describes interrelated things of interest in a specific domain of knowledge Creating a detailed ERD (Entity-Relationship Diagram) for an e-commerce system can be quite complex, as it depends on the specific requirements and features of the system. However, I can provide you with a simplified example to give you an idea of what an ERD for an e-commerce system might look like. Keep in mind that this is a basic representation, and a real-world system would likely have additional entities and relationships.



Flowchart:- A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence. They can range from simple, hand-drawn charts to comprehensive computer-drawn diagrams depicting multiple steps and routes.



# 5.Technologies to be used

## 5;1Software Platform

1. **Front-end**

* HTML5/CSS3: Description: HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) are fundamental technologies for structuring and styling web pages. HTML5 introduces new elements and APIs, while CSS3 provides advanced styling options.
* Usage: Used for creating the basic structure, layout, and styling of your e-commerce website.
* JavaScript (ES6+): Description: JavaScript is a versatile scripting language that enables dynamic content, interactivity, and improved user experience on the client side.
* Usage: Implement interactive features, such as product carousels, image sliders, form validation, and real-time updates without page reloads. Popular JavaScript frameworks and libraries include React, Angular, and Vue.js.
* React (or another JS framework/library): Description: React is a popular JavaScript library for building user interfaces. It is component-based, making it easier to manage and update UI components efficiently.
* Usage: Build dynamic and responsive user interfaces. React is commonly used for single-page applications (SPAs) in e-commerce.
* Bootstrap (or another CSS framework): Description: Bootstrap is a front-end framework that provides a set of pre-designed components and styles, allowing for rapid development and ensuring a consistent and responsive design. Usage: Use Bootstrap or a similar CSS framework to create a mobile-friendly, responsive design without starting from scratch.
* Graph QL or REST API : Description: Graph QL and REST are communication protocols that allow the front-end to interact with the back-end servers to fetch and update data.
* Usage: Use Graph QL or RESTful APIs to handle data exchange between the client and server. This is crucial for functionalities such as product listing, user authentication, and order processing.

1. **Back-end**

* Programming Language: Node.js: Known for its event-driven, non-blocking I/O model, Node.js is suitable for building scalable and high-performance applications. It allows developers to use JavaScript on both the client and server sides, promoting code reuse.
* Web Framework: Django (Python): Django is a high-level Python web framework that follows the "don't repeat yourself" (DRY) principle. It comes with built-in features for authentication, security, and database access, making it efficient for e-commerce development.
* Database Management System (DBMS): MySQL or PostgreSQL: These are popular open-source relational database management systems. They provide robust data management capabilities and are widely used for e-commerce applications due to their reliability, performance, and scalability.
* Server Environment: Express.js (Node.js): If you choose Node.js for your back-end, Express.js is a minimal and flexible Node.js web application framework. It simplifies the process of building scalable and maintainable web applications.
* Cloud Services : AWS (Amazon Web Services) : Leveraging cloud services is crucial for scalability and reliability. AWS and Azure provide a wide range of services, including compute power, storage, databases, and more, allowing you to build a scalable and robust e-commerce infrastructure**.**

## Hardware Platform

RAM, Hard Disk, OS, Editor, Browser etc.

|  |  |
| --- | --- |
| **Name Of Component** | **Specification** |
| Processor | Intel Core i5 |
| RAM | 16 Gb |
| HARD DISK | 500 Gb |
| OS | Window 10 |
| EDITOR | visual studio code |
| BROWSER | Google Chrome |

# 6Advantages of this Project

* Global Reach
* 24/7 Availability
* Cost Savings
* Data Collection and Analysis
* Personalization
* Efficient Inventory Management
* Streamlined Marketing
* Ease of Scaling
* Customer Engagement
* Reduced Transaction Costs

# 7.Future Scope and further enhancement of the Project

The future scope and enhancement of an e-commerce website project involve staying ahead of industry trends, adopting emerging technologies, and continuously improving user experience. Here are some potential areas for future development and enhancement:

**Mobile Commerce** :-Optimize the e-commerce platform for mobile devices and consider developing a dedicated mobile app. Mobile commerce is increasingly popular, and providing a seamless mobile experience can enhance user engagement.

**Progressive Web App (PWA)**:- Implementing PWA features can improve website performance, offline access, and provide a more app-like experience, enhancing user engagement and satisfaction.

**Voice Commerce**: - Integrate voice-activated shopping features to cater to the growing trend of voice-assisted devices. Voice commerce can simplify the shopping process for users who prefer hands-free interactions.

**Augmented Reality (AR) and Virtual Reality (VR)**:-Explore AR and VR technologies to enhance the online shopping experience. AR can be used for virtual try-on of products, while VR can create immersive shopping environments.

**Artificial Intelligence (AI) and Machine Learning (ML)**:-Implement AI and ML algorithms for personalized product recommendations, predictive analytics, and enhanced customer support through chat bot. These technologies can also be used for fraud detection and prevention..

**Subscription Services**:-Introduce subscription-based models or loyalty programs to encourage customer retention. Offering subscription boxes, exclusive discounts, or early access to products can create a sense of loyalty among customers.

**Social Commerce Integration**:-Leverage social media platforms for social commerce. Integrate social sharing features, user-generated content, and enable direct purchasing through social channels to reach a wider audience.

**Eco-Friendly Practices** :-Emphasize sustainability and eco-friendly practices. Implement features such as carbon footprint tracking, eco-friendly product labels, and sustainable packaging options to appeal to environmentally conscious consumers.

**Cross-Border E-Commerce**:-Explore opportunities for international expansion and cross-border e-commerce. Provide support for multiple currencies, languages, and shipping options to cater to a global customer base.

# 8.Team Details

| **Project Name & ID** | **Course Name** | **Student ID** | | **Student Name** | **Role** | **Signature** |
| --- | --- | --- | --- | --- | --- | --- |
| Hosting E-Commerce Website in AWS | Industrial Training | TCA2211011 | | Harsh Madhur | Developer |  |
|  | |  |  |  |
| Hosting E-Commerce Website in AWS | Industrial Training | TCA2211002 | Aman Kumar | | Testing |  |
|  |  | |  |  |

# 9.Conclusion

The completion of our e-commerce project marks a significant milestone, and we reflect on the journey with a sense of accomplishment and learning. Throughout the development process, we aimed to create a robust, user-friendly platform that aligns with current industry trends and meets the evolving needs of our users.

In conclusion, the successful completion of the e-commerce project represents a significant step forward in our digital journey. We are committed to staying agile, adapting to market changes, and continually enhancing the platform to meet the expectations of our users and stakeholders.

We extend our gratitude to the project team, stakeholders, and users for their collaboration, feedback, and support throughout this endeavor. As we look to the future, we remain dedicated to the ongoing success and evolution of our e-commerce platform.

Thank you.

# 10.References

* <https://en.wikipedia.org/wiki/E-commerce>
* <https://aws.amazon.com/what-is-aws/>
* <https://www.ecommercetimes.com/>
* <https://www.youtube.com/watch?v=xuAltdNpvC0&pp=ygUJZWNvbW1lcmNl>
* <https://www.aws.training/>