



MCA 4<sup>TH</sup> SEMESTER (2023-2025)

**ROURKELA INSTITUTE OF MANAGEMENT STUDIES**

(Affiliated to Biju Patnaik University of Technology)



**Rourkela Institute of Management Studies**

Rourkela

Department of Computer Science Rourkela Institute of Management Studies Chhend,

Rourkela-15, Odisha Phone:0661 2480482

Fax: 91-0661-1480665

Mail: [rkl\\_rimsgrol@sancharnet.in](mailto:rkl_rimsgrol@sancharnet.in) Visit: [www.rims-edu.com](http://www.rims-edu.com)

## **CERTIFICATE OF EXAMINATON**

This is to certify that this project report entitled "**FashIn**" submitted by **Aditya Khandual** of 4<sup>th</sup> semester, **Rourkela Institute of Management Studies, Rourkela**, is accepted as partial fulfillment of requirements for the degree in Master in Computer Applications, under **Biju Patnaik University of Technology, Rourkela** this has been verified by us and found be original up to our satisfaction.

**Examiner**



**Rourkela Institute of Management Studies**

Rourkela

Department of Computer Science Rourkela Institute of Management Studies, Chhend,

Rourkela-15, Odisha Phone:0661 2480482

Fax:91-0661-1480665

Mail: [rkl\\_rimsgrol@sancharnet.in](mailto:rkl_rimsgrol@sancharnet.in) Visit: [www.rims-edu.com](http://www.rims-edu.com)

**CERTIFICATE OF EXAMINATON**

This is to certify that this project report entitled “**FashIn**” submitted by **Aditya Khandual** of 4<sup>th</sup> semester, **Rourkela Institute of Management Studies, Rourkela**, is accepted as partial fulfilment of requirements for the degree in Master In Computer Applications, under **Biju Patnaik University of Technology, Rourkela** this has been verified by us and found be original up to our satisfaction.

**Examiner**



**Rourkela Institute of Management  
Studies**

Rourkela

Department of Computer Science Rourkela

Institute of Management Studies Chhend,

Rourkela-15, Odisha Phone:0661 2480482

Fax:91-0661-1480665

Mail: [rkl\\_rimsgrol@sancharnet.in](mailto:rkl_rimsgrol@sancharnet.in) Visit:

[www.rims-edu.com](http://www.rims-edu.com)

**CERTIFICATE**

This is to certify that this project entitled "**FashIn**" has been and submitted by **Aditya Khandual** M.C.A 2023-2025, **Rourkela Institute of Management Studies, Rourkela**, has been examined by us.

He is found fit and approved for the award of "**Master in Computer Application** "Degree.

To the best my knowledge this work has not been submitted for the award of any other degree.

I wish all success in his life.

**DEAN ACADEMICRIMS, ROURKELA**



Prof. Bibhudendu Panda Head of The  
Department, MCA

Rourkela Institute of Management Studies  
Rourkela

### **CERTIFICATE**

This is to certify that **Aditya Khandual** student of **M.C.A,**  
**Rourkela Institute of Management Studies, Rourkela, Odisha** of Session 2023-  
2025 has completed the project successfully.

I wish all success in his life

**(Prof. Bibhudendu Panda)**



### **DECLARATION**

I, **Aditya Khandual**, hereby declare that the project report entitled "**FashIn**" is of my work. The above work I submitted to "**Biju Patnaik University of Technology, Rourkela**" for the award of "**Master in Computer Applications**" Degree.

To the best of my knowledge, this work has not been submitted or published anywhere for the award of any degree.

Aditya Khandual



## ACKNOWLEDGEMENT

I would like to express my gratitude to **Mr. Rashmi Kant Das (JavaTechnocrat, BBSR)** and **Mr. Smruti Ranjan Nayak** for his guidance and support during the project work.

I am deeply indebted to **Rourkela Institute of Management Studies, Chhend, Rourkela**, for providing me an opportunity to undertake a project work entitled "**FashIn**".

I am grateful to my project guide **Prof. Bibhudendu Panda** without his guidance it would not have been possible on my part to complete the project.

I acknowledge the help and co-operation received from all my team members in making this project.

I consider myself fortunate that I have successfully completed this project; I acknowledge my sincere gratitude to all those works and ideas that had helped me in writing this project.

*Aditya Khandual*

**University Roll No: 2305260002**

**MCA (2023-2025)**

**Rourkela Institute of Management Studies,  
Rourkela.**

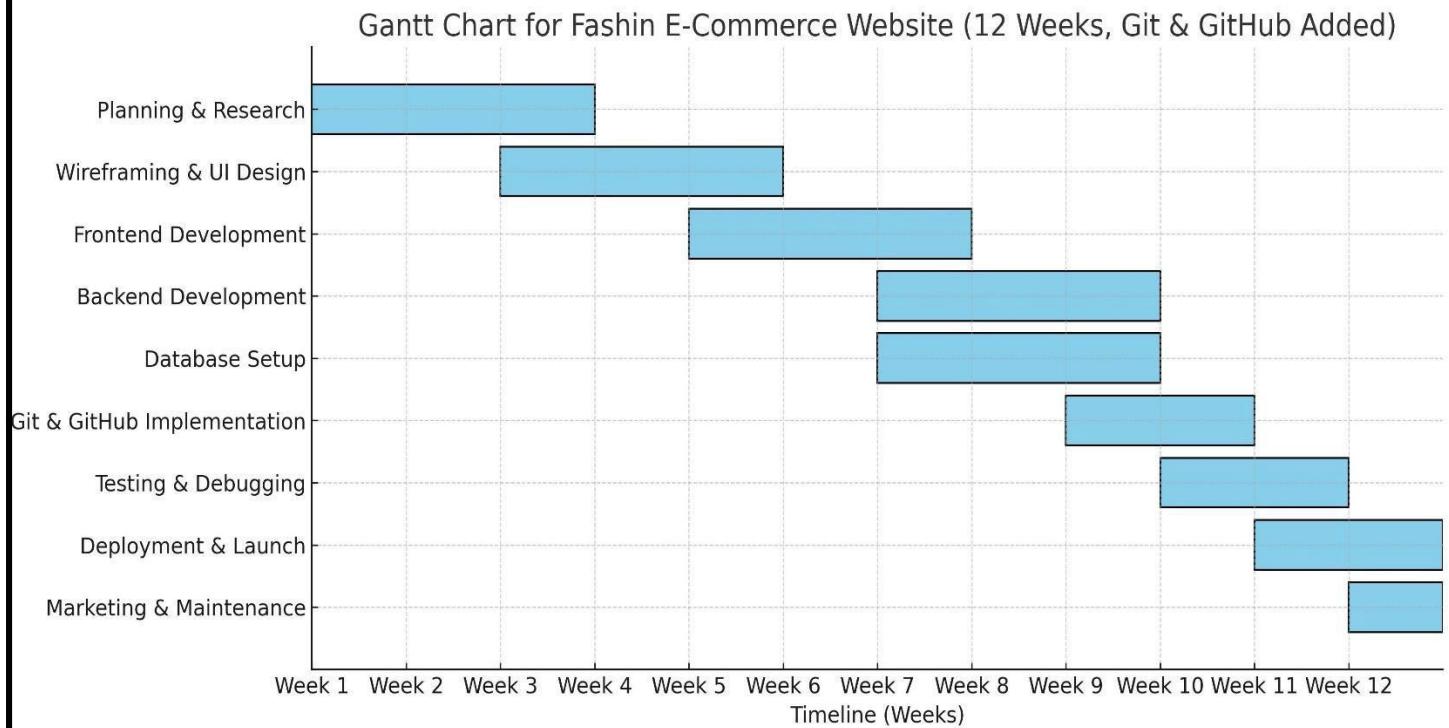
## **ABSTRACT**

FashIn is a dynamic and user-friendly e-commerce web application designed to offer a seamless shopping experience for fashion and lifestyle products. Built using **Spring Boot**, FashIn provides a robust and scalable backend that ensures smooth functionality, secure transactions, and efficient data management. The application features an intuitive user interface, allowing customers to browse, search, and purchase a wide range of fashion items with ease.

Key functionalities include **user authentication, product catalogue management with vast categories option**. The backend, powered by **Spring Boot**, is integrated with a database to manage users, products, and orders efficiently. Additionally, the application follows **MVC architecture** and utilizes **Thymeleaf** for dynamic content rendering.

FashIn aims to enhance the online shopping experience with a responsive design, smooth navigation, and a secure checkout process. By leveraging the power of **Spring Boot**, this e-commerce platform ensures high performance, security, and scalability for future growth.

# GANNT CHART



## Table of Contents

<u>Sl.No.</u>	<u>Title</u>	<u>Page No.</u>
1	FRONT PAGE	01
2	CERTIFICATION OF EXAMINATION	02-03
3	CERTIFICATE	04-05
4	DECLARATION	06
5	ACKNOWLEDGEMENT	07
6	ABSTRACT	08
7	GANTT CHART	09
8	INTRODUCTION	11-15
9	LITERATURE SURVEY	15
10	PROJECT PLANNING	16-17
11	ENVIRONMENT SETTINGS	18-33
12	SELECTED SOFTWARE PLATFORM	34-37
13	DATABASE WITH SNAPSHOTS	38-49
14	SOFTWARE REQUIRE SPECIFICATION	50-51
15	DESIGN	52-55
16	CODING WITH SNAPSHOTS	56-75
17	TESTING	76-77
18	REFERENCES	78
19	CONCLUSION AND FUTURE SCOPE	79

# CHAPTER-1

## INTRODUCTION

### 1.1 Preface

The Fashin E-Commerce Website is designed to provide a seamless and stylish online shopping experience focused on fashion and lifestyle products. This project plan outlines the various stages involved in developing the website, from initial research to final deployment. The structured timeline ensures an organized workflow, covering frontend and backend development, database setup, and essential integrations like Git and GitHub for version control.

This document presents a Gantt chart that visually represents the tasks and their respective timelines, helping track progress and ensuring that the project stays on schedule. With a strong foundation in design, functionality, and user experience, Fashin aims to establish itself as a reliable and engaging online shopping platform.

### 2. Understanding the Problem Statement

With the rise of online shopping, consumers expect a seamless and user-friendly platform to browse, purchase, and engage with fashion and lifestyle products. The primary challenge is to design an e-commerce website that not only meets these expectations but also provides a visually appealing interface, efficient navigation.

Key problems to address include:

- Creating a responsive and aesthetically pleasing design that attracts customers.
- Ensuring smooth frontend and backend interactions for an optimal shopping experience.
- Implementing a robust database structure to manage products, users, and orders efficiently.
- Utilizing Git and GitHub for version control and collaborative development.

By addressing these challenges, Fashin aims to deliver a high-quality shopping experience that enhances customer satisfaction and engagement.

### 3. Motivation for Work

#### Motivation for Work

The motivation behind developing the Fashin E-Commerce Website stems from the growing demand for convenient and stylish online shopping experiences. With the rapid evolution of fashion trends and increasing reliance on digital platforms, it has become essential to create a website that not only showcases products but also offers an engaging and hassle-free shopping journey.

Key motivations include:

- Bridging the gap between fashion brands and consumers through a dynamic online platform.
- Providing a well-structured and responsive design to enhance user interaction.
- Establishing a scalable and maintainable solution for long-term business growth.

### **3. Project Overview**

The Fashin E-Commerce Website is a comprehensive online shopping platform focused on delivering high-quality fashion and lifestyle products. The project encompasses various aspects of web development, ensuring a feature-rich and user-friendly experience.

#### **Key Objectives:**

- Develop a visually appealing and responsive website layout for seamless browsing.
- Implement a robust backend system for efficient order processing and user management.
- Establish a secure and scalable database structure to store product and customer information.
- Utilize Git and GitHub for effective version control and team collaboration.
- Conduct rigorous testing to enhance security, performance, and usability.

#### **Core Features:**

- **User Authentication** – Secure login and registration system.
- **Product Catalogue** – Well-organized display of fashion and lifestyle products.
- **Responsive Design** – Mobile-friendly interface for an optimized shopping experience.
- **Admin Panel** – Tools for managing inventory, users, and orders.

With a structured development approach, Fashin aims to offer an innovative and engaging online shopping platform that meets modern user expectations.

## 4. Purpose

The purpose of developing the Fashin E-Commerce Website is to create an intuitive, engaging, and efficient platform that caters to modern online shopping needs.

### Primary Purposes:

- **Enhancing User Experience:** Providing a seamless, responsive, and visually appealing shopping experience.
- **Increasing Business Reach:** Helping fashion and lifestyle brands expand their customer base through an online marketplace.
- **Leveraging Technology for Growth:** Utilizing modern web development tools, databases, and version control systems like Git and GitHub to ensure a scalable and maintainable solution.
- **Facilitating Inventory & Order Management:** Offering an easy-to-use backend system for managing products, customers.

By focusing on these purposes, Fashin aims to redefine the fashion e-commerce experience while providing a sustainable and growth-oriented business model.

## Project Scope

The scope of the Fashin E-Commerce Website encompasses all the essential aspects of development, design, implementation, and maintenance to ensure a functional and successful online shopping platform.

### In-Scope:

#### • Website Development:

- Frontend design using HTML, CSS, JavaScript (including responsive design elements).
- Backend development using SpringBoot and MySQL.
- Database integration for efficient product and user management.

#### • User Features:

- Secure authentication (login, signup).
- User-friendly product browsing and filtering.
- Shopping cart, checkout functionalities.

#### • Admin Features:

- Product and inventory management.
- Order and user management dashboard.

#### • Security Measures:

- Data encryption and user authentication protocols.

#### • Version Control & Collaboration:

- Git and GitHub implementation for tracking changes and team collaboration.

#### • Testing & Deployment:

- Unit testing, performance testing.

## 5. Project Goals and Objectives

### Project Goals:

- 1. Create a User-Friendly E-Commerce Platform** – Develop an intuitive, easy-to-navigate website that offers a seamless shopping experience.
- 2. Offer a Wide Range of Fashion & Lifestyle Products** – Provide users with a curated collection of clothing, accessories, and lifestyle products.
- 3. Ensure a Responsive and Modern Design** – Make the website visually appealing and fully functional across all devices (desktop, tablet, and mobile).
- 4. Optimize Performance & Speed** – Build a fast-loading platform to reduce bounce

### Project Objectives:

- 1. Develop the Core Website Structure** – Build a homepage, product listing pages, product details pages, and a user dashboard.
- 2. Integrate a Shopping Cart and Checkout System** – Allow users to add items to their cart and complete purchases seamlessly.
- 3. Enable User Registration & Profile Management** – Provide customers with account creation.
- 4. Implement Search** – Help users find products easily using categories, and search bars.
- 5. Ensure Mobile Optimization & Cross-Browser Compatibility** – Ensure the website works smoothly on different devices and browsers.
- 6. Set Up Admin Dashboard for Product & Order Management** – Provide an easy-to-use admin panel for managing inventory, orders, and customer queries.

### Requirements Specifications

➤ RAM-4 Gb or Higher

➤ HDD-1TB

➤ SSD-120Gb

➤ Processor-i3 or Higher

### Software Specifications

➤ Windows- win-7, win-10, win-11

➤ Frontend Language- HTML, CSS, JAVASCRIPT

➤ Backend Language- Java

- Database: MySql
- Framework: SpringBoot
- Source Controller- Git & Github

## CHAPTER-2

---

### LITERATURE SURVEY

---

#### Literature Survey on E-commerce in Fashion and Lifestyle

#### 1. Introduction

E-commerce has revolutionized the fashion and lifestyle industry by providing convenient, accessible, and personalized shopping experiences. With the rise of digital transformation, online platforms have become the primary mode of purchasing fashion products, enabling consumers to shop from anywhere at any time. This literature survey explores the key aspects of e-commerce in the fashion and lifestyle industry, highlighting trends, challenges, and technological advancements.

#### 2. Evolution of E-commerce in Fashion and Lifestyle

The fashion industry has transitioned from traditional brick-and-mortar stores to online platforms. Early e-commerce platforms such as Amazon and eBay paved the way for specialized fashion retailers like ASOS, Zara, and H&M to establish their digital presence. The emergence of mobile commerce, social commerce, and omnichannel retailing has further enhanced the online shopping experience.

#### 3. Consumer Behavior in Online Fashion Shopping

Studies indicate that factors like convenience, price comparison, reviews, and personalized recommendations influence consumer purchasing decisions. Social media platforms such as Instagram and TikTok play a significant role in shaping fashion trends and driving sales through influencer marketing and direct shopping features.

# CHAPTER-3

## PROJECT PLANNING

### 3.1 Members and Capabilities

3.1: Table of Capabilities

SR. No	Name of Member	Capabilities
1	Simran Nayak	Backend and Security
2	Aditya Khandual	UI Design

### Work Breakdown Structure

- a) All of the members are equally important in developing the project.
- b) We work on a different part of the project based on one's capability.
- c) Firstly, we came up with documentation, and based on the documentation we set our goal and created a blueprint.
- d) We then started going hands-on with the project to develop it according to the flow as decided earlier

### 3.2 Roles and Responsibilities

Table 3.2: Table of Responsibilities

SR. No	Name of Member	Role	Responsibilities
1	Simran Nayak	Member	Backend & Security
2	Aditya Khandual	Member	HTML, CSS, JS

## Assumptions

- Consumers are willing to shift towards online shopping for fashion and lifestyle products.
- Internet accessibility and smartphone penetration will continue to increase, making e-commerce more accessible.
- Social media and influencer marketing will remain crucial in driving fashion trends and sales.
- Consumers will prefer a mix of affordability, sustainability, and personalization when shopping for fashion items.

## Constraints

- **Logistical Challenges:** Delivery delays, high shipping costs, and issues with international shipping.
- **Consumer Trust Issues:** Customers may hesitate to purchase due to concerns about product quality and fraud.
- **Regulatory Compliance:** Fashion e-commerce businesses must comply with data protection laws, taxation policies, and sustainability regulations.
- **Sustainability Limitations:** Ethical sourcing and eco-friendly practices may increase production costs.

### **3. Project Management Approach**

- a) Planning of project
- b) Defining the scope of the project.
- c) Estimation of time and It's management.
- d) Creating Gantt Charts and properly assigning tasks to members
- e) Reporting the progress of project with the guide.

### **4. Ground Rules for the Project**

- a) Properly planning and gathering relevant information is very important.
- b) Developing a Blueprint of the project and work accordingly.
- c) All the members should report to the guide whenever required
- d) Setting up small goals every week.
- e) Achieving the small goal within that span of time.
- f) Keeping tracks of the progress towards project.

### **5. Project Budget**

- a. It is moderate project with high end technology.
- b. Cost of the project is low and efficient.
- c. Time-consuming as it is on latest technologies.

# CHAPTER-4

---

## ENVIRONMENT SETTINGS

---

### Prerequisite

- Java (jdk.17)
- JavaScript (ecmaScript 6)
- HTML, CSS, JS(es6)
- SpringBoot
- Thymeleaf
- Microsoft Visual Studio Code
- Git and GitHub
- SpringBoot Dependencies:
  1. SpringWeb
  2. Spring JPA
  3. Spring Security
  4. Spring DevTool
  5. Validation
  6. Thymeleaf
  7. Mysql Driver
  8. Lombok

Developed any Project need a IDLE (Integrated Development Environment), so Developer can easily Create, Run, Compile and Execute Programs.

## How to Install Visual Studio Code on Windows?

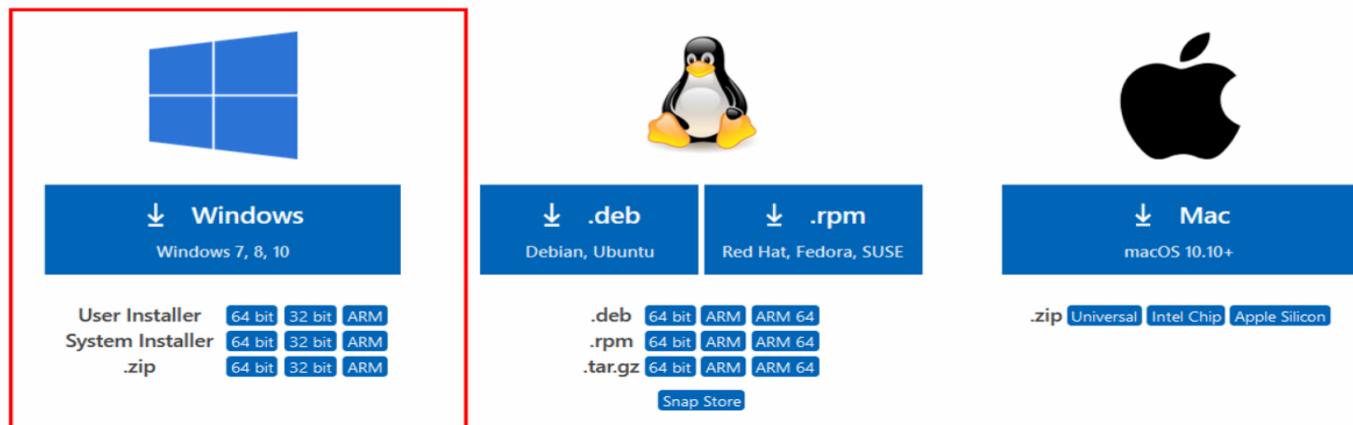
Firstly, download the Visual Studio Code installer for Windows. Once it is downloaded, run the installer (*VSCodeUserSetup-{version}.exe*). It will only take a minute.

Visit: <https://code.visualstudio.com/download>

## Download Visual Studio Code

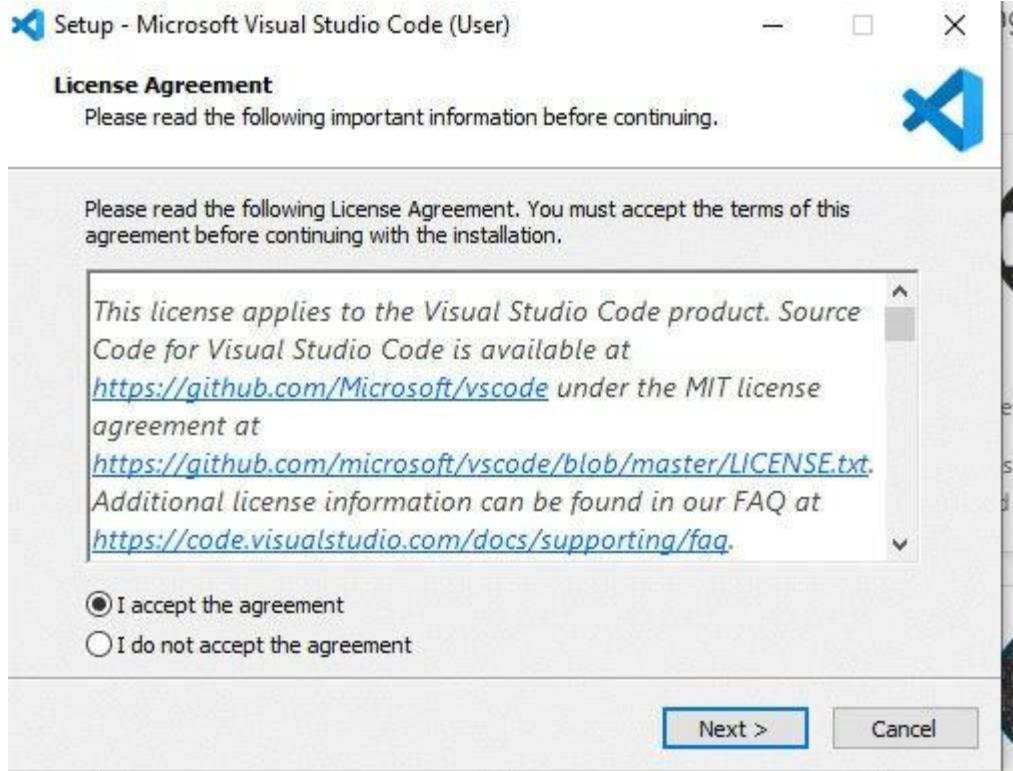
Free and built on open source. Integrated Git, debugging and extensions.

### How to Install Visual Studio Code on Windows and Run Python Programs

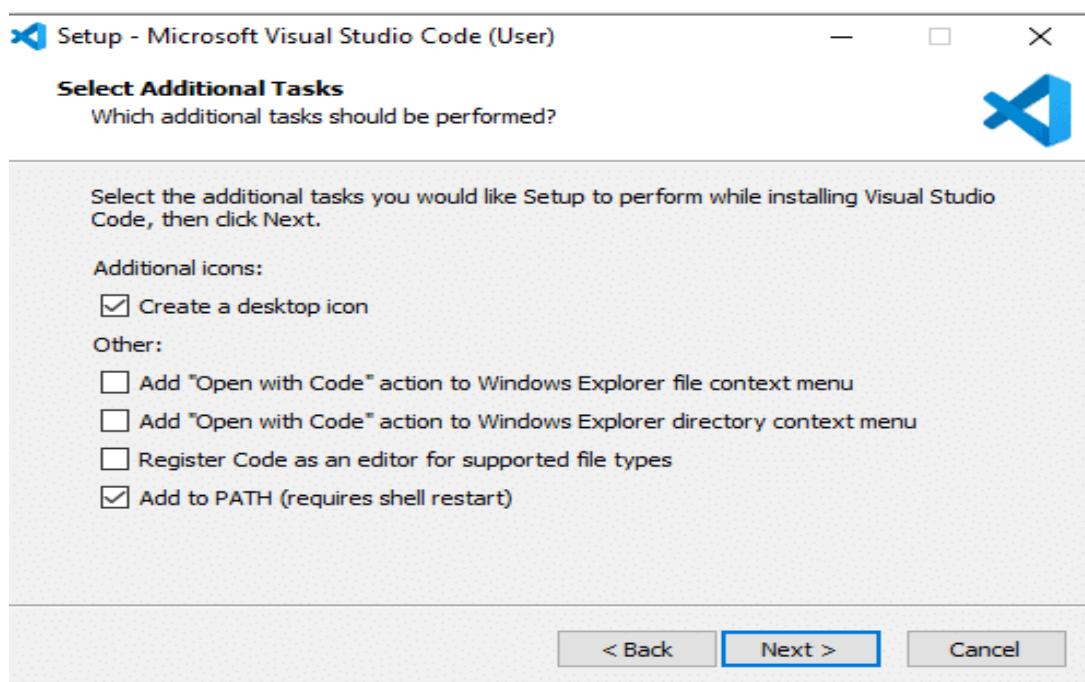


Then double Click on Download folder .exe file and Follow the step by step instruction Process .

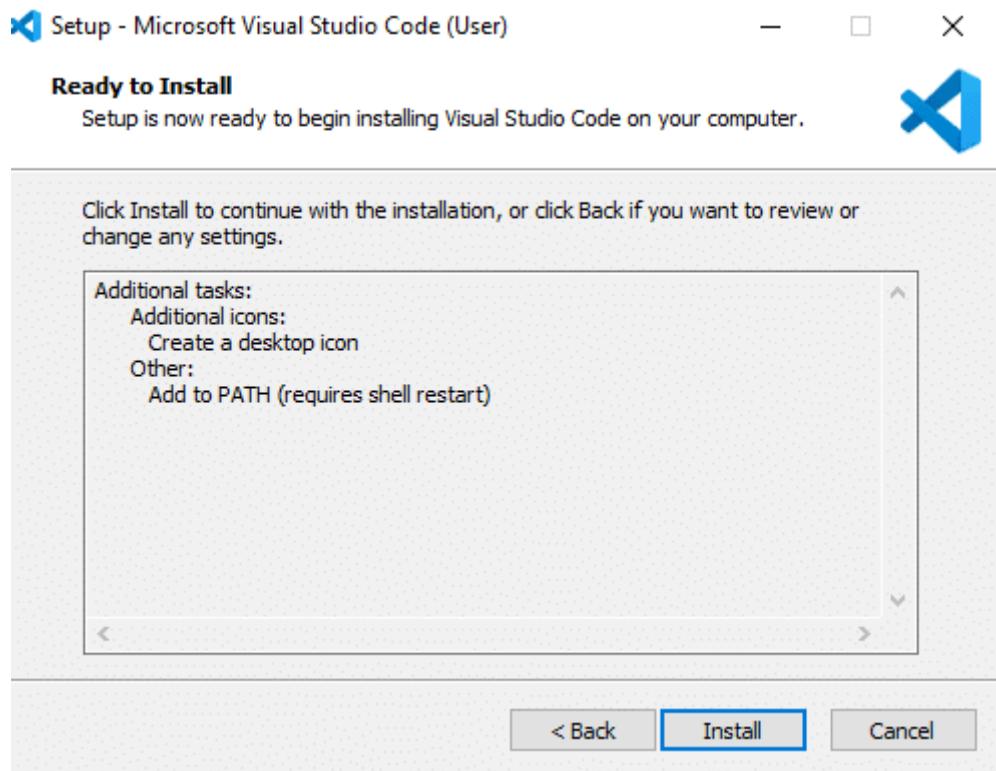
Secondly, accept the agreement and click on next.



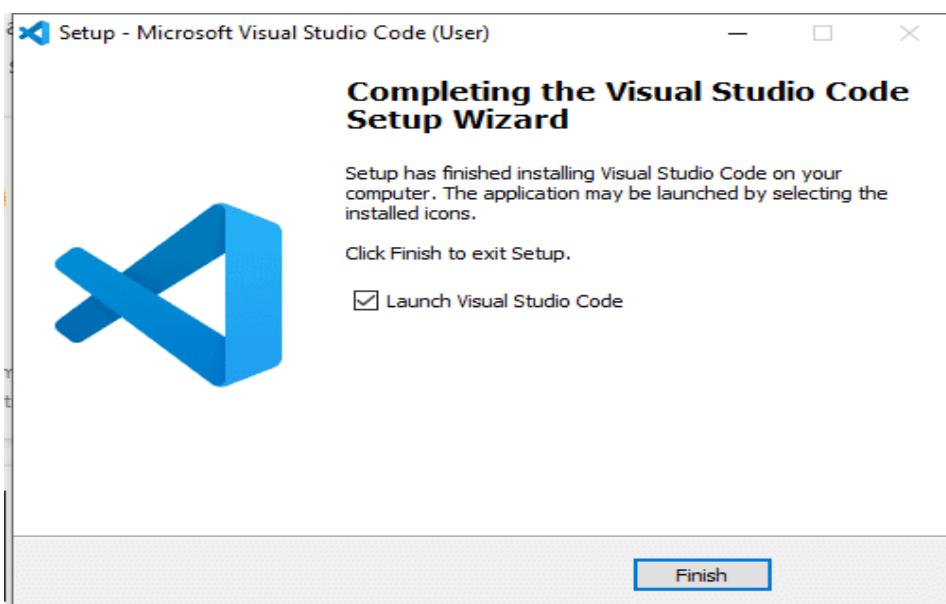
Thirdly, click on “create a desktop icon” so that it can be accessed from desktop and click on Next.



After that, click on the install button.



Finally, after installation completes, click on the finish button, and the visual studio code will get open.



## How to initiate SpringBoot by adding Dependencies

### Method 1: Using Spring Initializr (Recommended)

#### 1. Go to Spring Initializr

Open <https://start.spring.io/>

The screenshot shows the Spring Initializr interface with the following configuration:

- Project:** Maven (selected)
- Language:** Java (selected)
- Spring Boot:** 3.4.4 (selected)
- Dependencies:** No dependency selected.
- Project Metadata:**
  - Group: edu-rims
  - Artifact: Fashln
  - Name: Fashln
  - Description: Demo project for Spring Boot
  - Package name: edu-rims.Fashln
  - Packaging: Jar (selected)
  - Java: 17 (selected)
- Buttons at the bottom:** GENERATE (CTRL + ⌘), EXPLORE (CTRL + SPACE), and a three-dot menu button.

## 2. Select Project Type & Language

- **Project:** Maven (or Gradle if you prefer)
- **Language:** Java
- **Spring Boot Version:** Latest stable version (e.g., 3.x.x)

## 3. Fill in Project Metadata

- **Group:** edu-rims (or your package name)
- **Artifact:** FashIn (name of your project)
- **Name:** FashIn
- **Package Name:** Project for Spring Boot
- **Packaging:** Zip
- **Java Version:** 17 (or as per your system compatibility)

## 4. Add Dependencies

Click on "Add Dependencies" and search for each dependency:

- **Spring Web** (For building REST APIs and web applications)
- **Spring Data JPA** (For database interaction using JPA)
- **Spring Security** (For authentication & authorization)
- **Spring Boot DevTools** (For live reload & development support)
- **Spring Boot Validation** (For data validation)
- **Thymeleaf** (For frontend templating)
- **MySQL Driver** (For database connectivity)
- **Lombok** (For reducing boilerplate code like getters & setters)



**Project**  
 Gradle - Groovy    Gradle - Kotlin  
 Maven

**Language**  
 Java    Kotlin    Groovy

**Dependencies**

**Spring Web** WEB  
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

**Spring Boot DevTools** DEVELOPER TOOLS  
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

**Spring Data JPA** SQL  
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

**Spring Security** SECURITY  
Highly customizable authentication and access-control framework for Spring applications.

**Lombok** DEVELOPER TOOLS  
Java annotation library which helps to reduce boilerplate code.

**MySQL Driver** SQL  
MySQL JDBC driver.

**Thymeleaf** TEMPLATE ENGINES  
A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

**Validation** I/O  
Bean Validation with Hibernate validator.

**Project Metadata**

Group	edu-rims
Artifact	FashIn
Name	FashIn
Description	Demo project for Spring Boot
Package name	edu-rims.FashIn
Packaging	<input checked="" type="radio"/> Jar <input type="radio"/> War
Java	<input type="radio"/> 24 <input type="radio"/> 21 <input checked="" type="radio"/> 17

**GENERATE** CTRL + D   **EXPLORE** CTRL + SPACE   ...

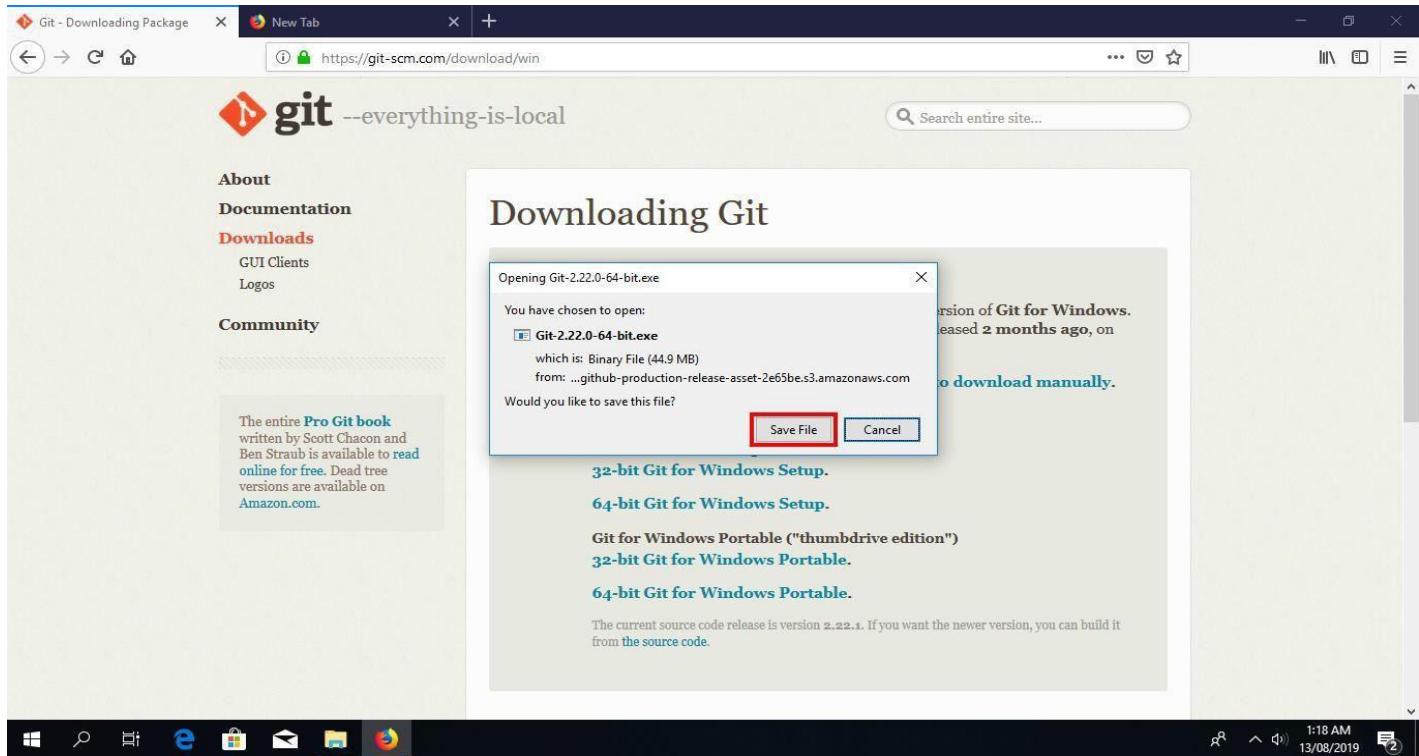
## 5. Generate the Project

- Click on **Generate**
- It will download a **zip file**
- Extract the zip and open the project in **IntelliJ IDEA, Eclipse, or VS Code**

# Install Git

## 1. Download Git

- Go to the [official Git website](#).
- Download the latest version for your operating system (Windows, macOS, or Linux).

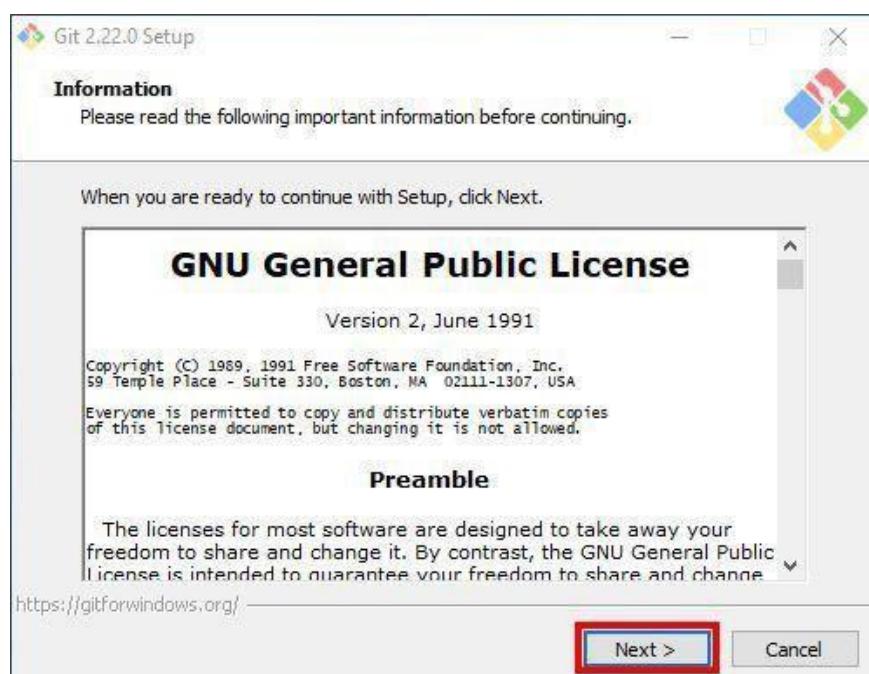


Click on “Save File” to start downloading the executable.

## 2. Install Git Bash

Run the Installer

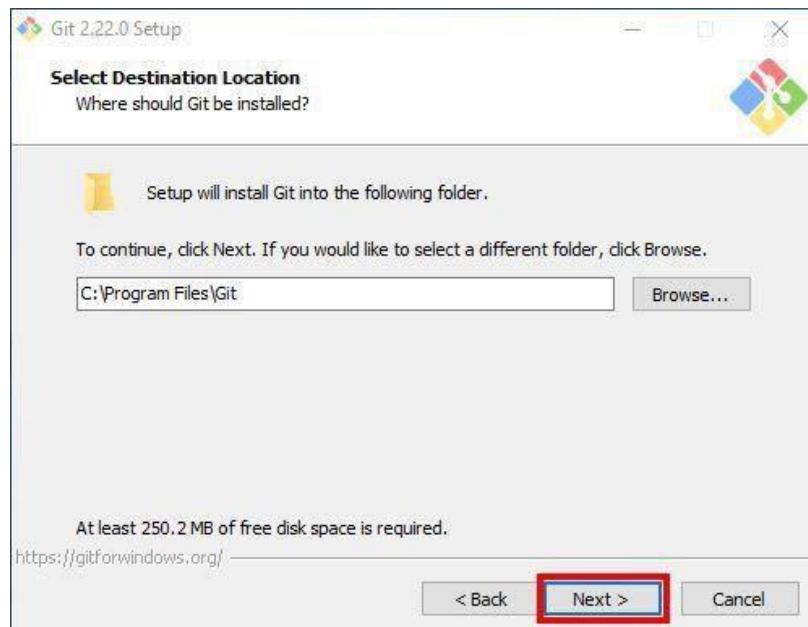
Once you have downloaded the Git Click “Next” after you have read the license Bash executable,



Click “Next” after you have read the license.

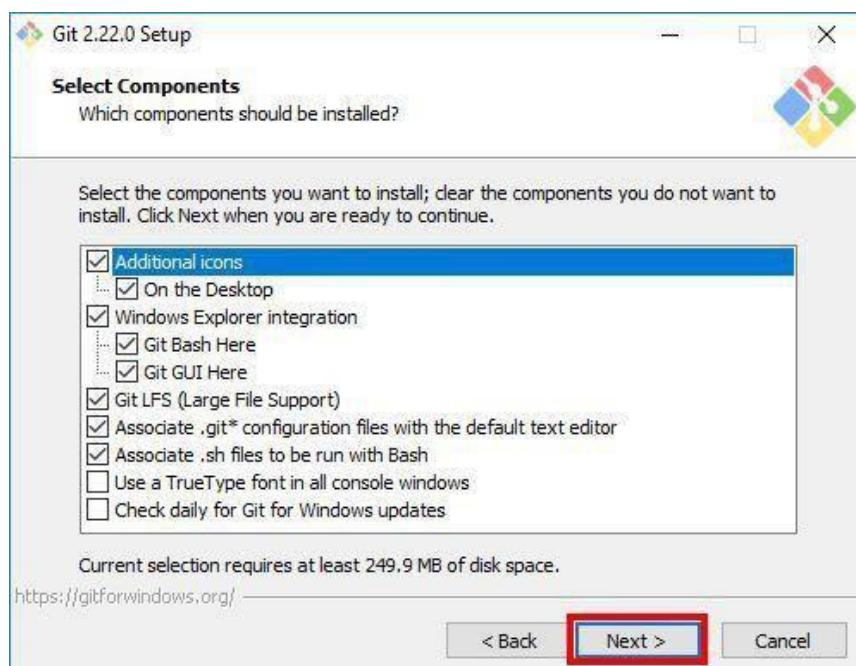
## Step 4: Select Destination Location Permalink

Next, select the location you want to install Git Bash. I would recommend you just leave the default option as it is, and click “Next”.



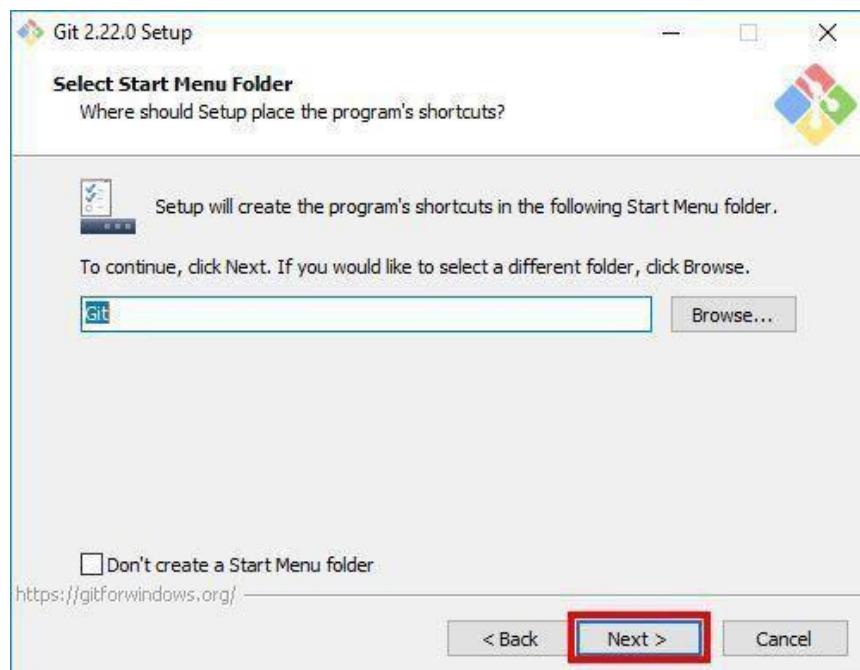
## Step 5: Select Components

Choose the components you want to install, or you can just proceed with the default options and click “Next”. I prefer selecting the “Additional icons” component which creates a Git Bash shortcut on the desktop.



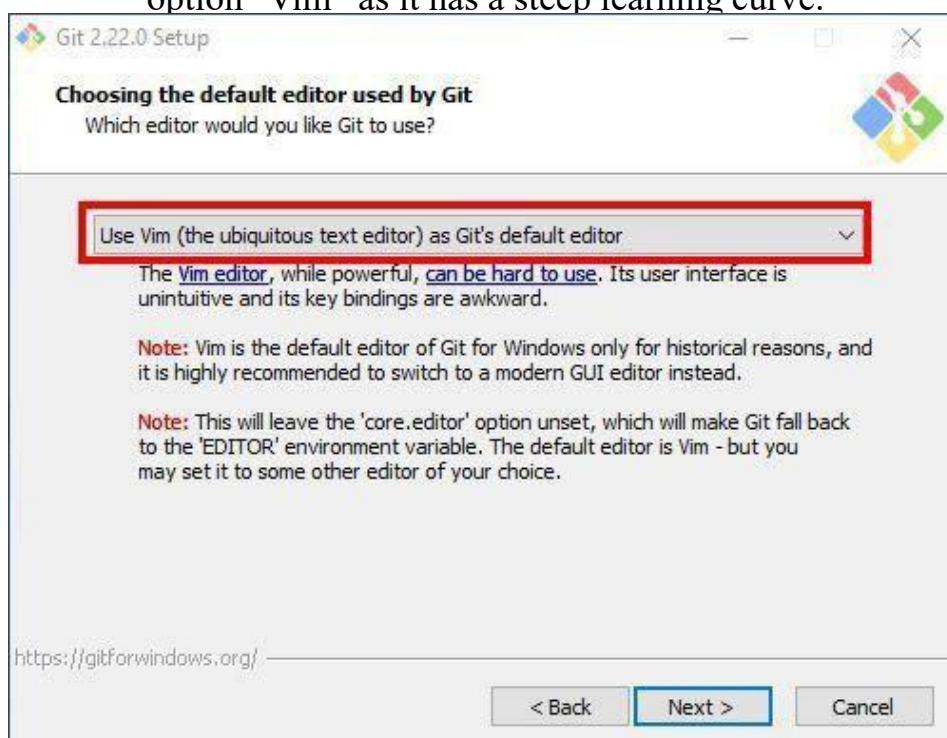
## Step 6: Select Start Menu Folder

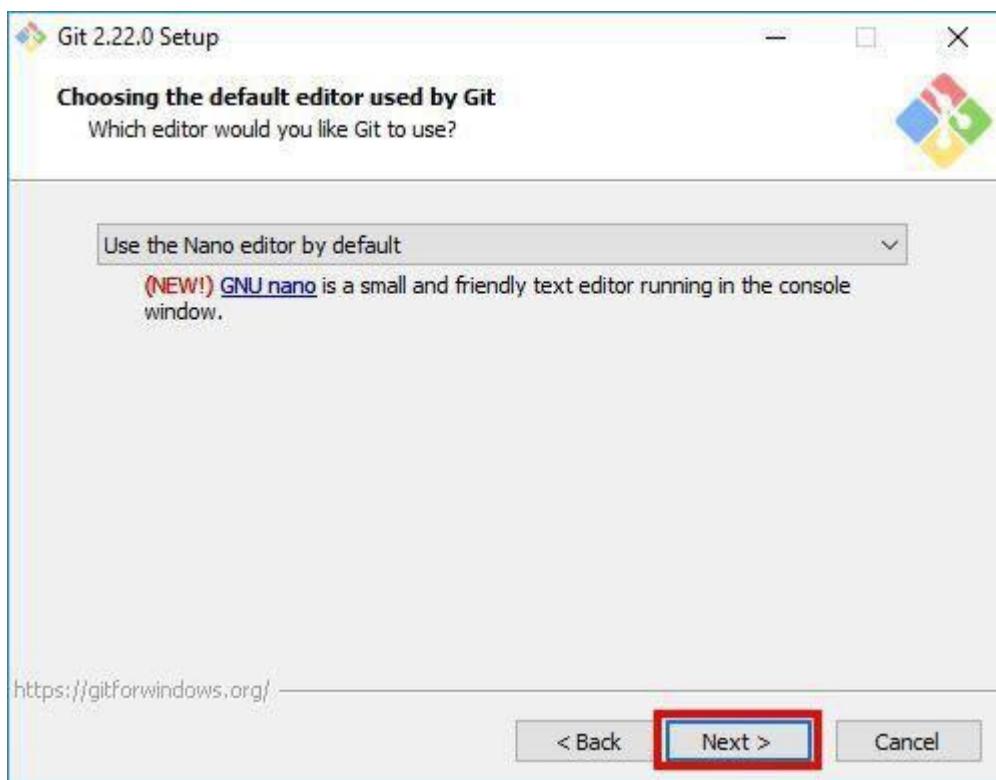
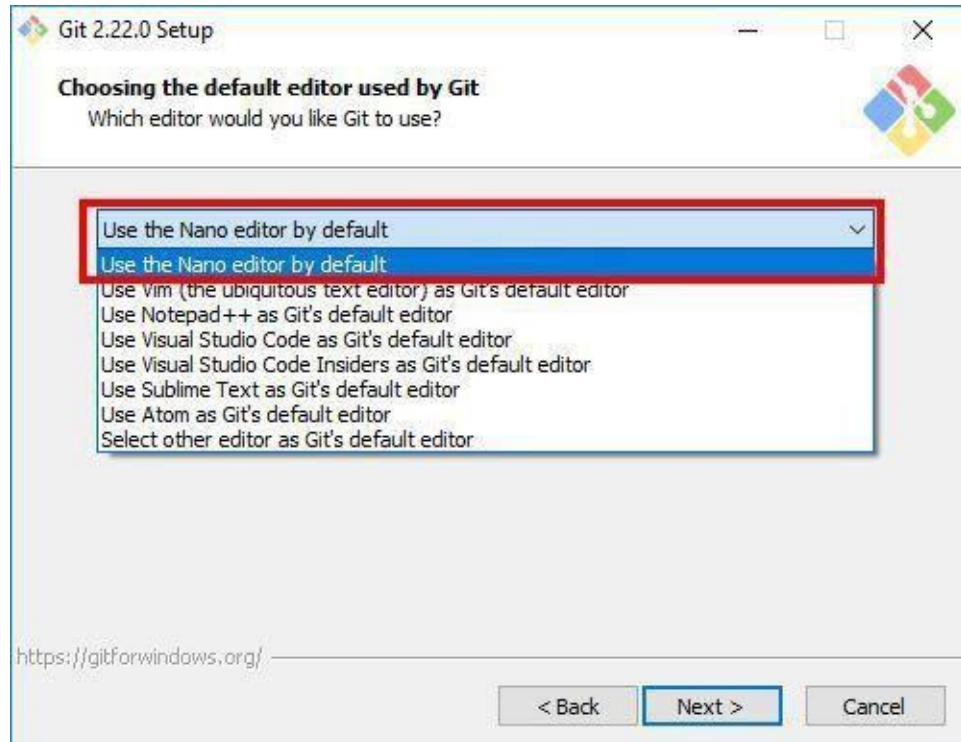
You can change the name of start menu folder here if you want, or just leave the default name and click “Next”.



## Step 7: Choose the Default Editor used by Git

Next, select the default editor for Git to use. Choose the one you like and click “Next”. I would recommend you proceed with **Nano** or **Notepad++**. Don’t proceed with the default option “Vim” as it has a steep learning curve.





## Step 8: Adjust your PATH Environment

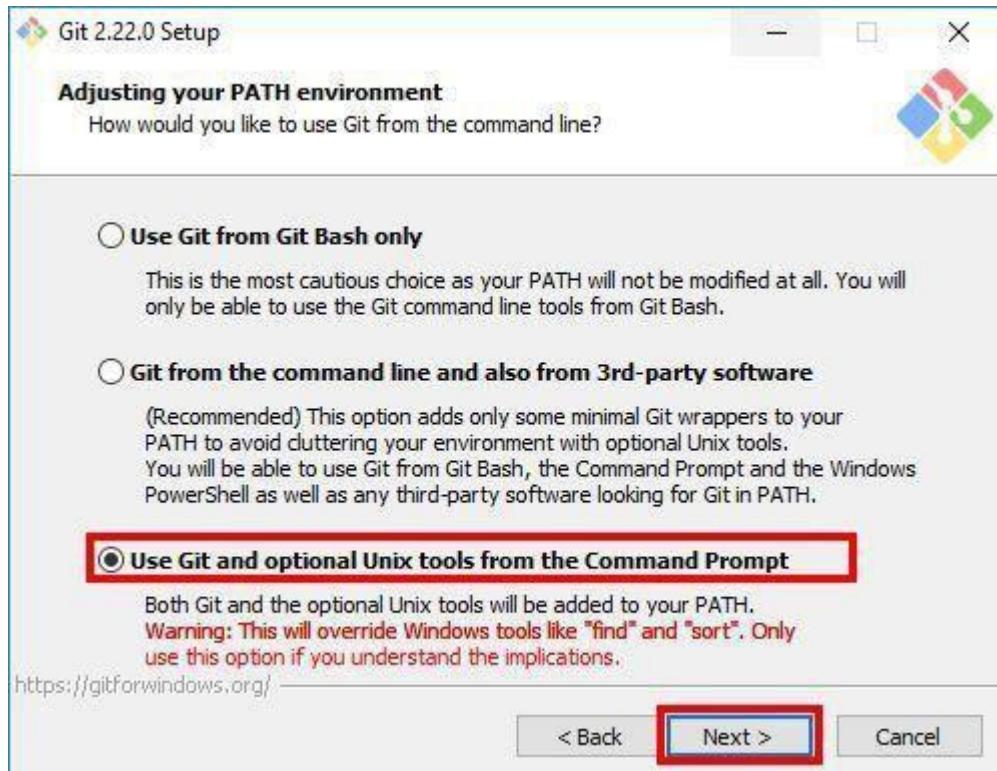
Choose the option you want depending on where you want to use Git and click “Next”. Select “**Use Git from Git Bash only**” option if want to run Git and Bash commands from Git Bash only. This means that you won’t be able to run Git commands such as `git status` on Windows Command Prompt or Powershell. They will only be found on Git Bash.

Select “**Git from the command line and also from 3rd-party software**” option if you want to run Git commands on Windows Command Prompt or Powershell.

**Notice:** Bash commands won't work on Command Prompt or Powershell with this option, but only Git commands will work.

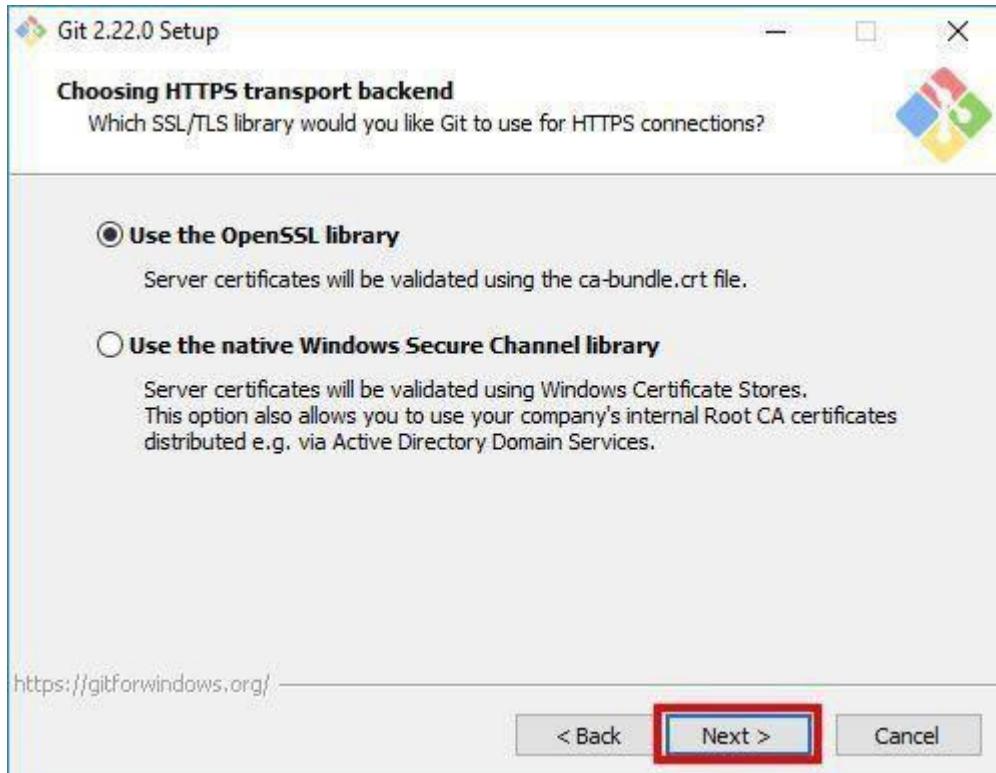
**Tip:** If you need run bash commands, you will have to open the Git Bash. So go ahead with this option if that is what you want.

Select “**Use Git and optional Unix tools from the Command Prompt**” option if you want to use both Git and Bash commands on Windows Command Prompt or Powershell. This option will override some default Windows Command Prompt tools like find and sort. I don’t use CMD or Powershell that much to worry about that. So I will go ahead with this option by clicking “Next”.



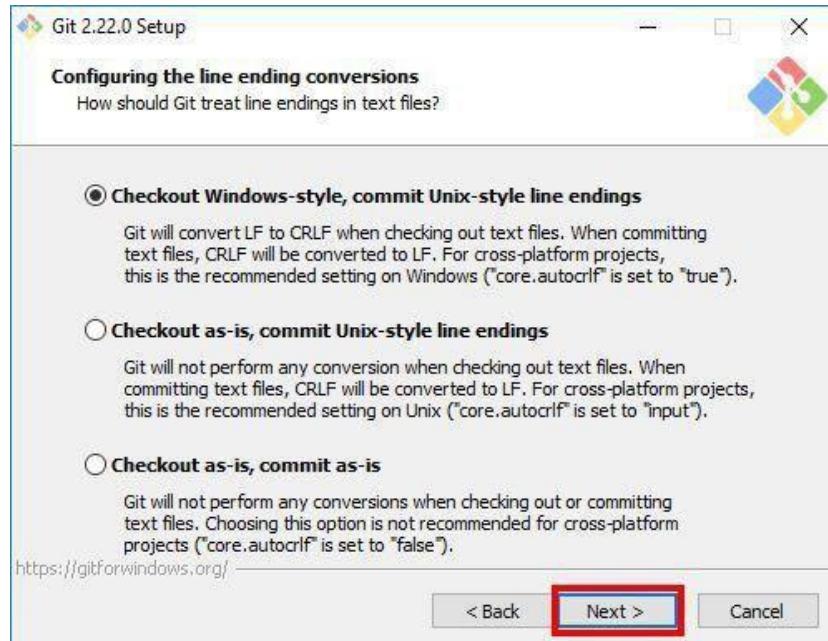
#### Step 9: Choose HTTPS Transport BackendPermalink

Next, select “Use the OpenSSL library” and click “Next”.



## Step 10: Configure the Line Ending Conversions

Select how Git should treat line endings in text files. It's probably safe to go with the default option "Checkout Windows-Style, commit Unix-style line endings". Click "Next" to proceed.



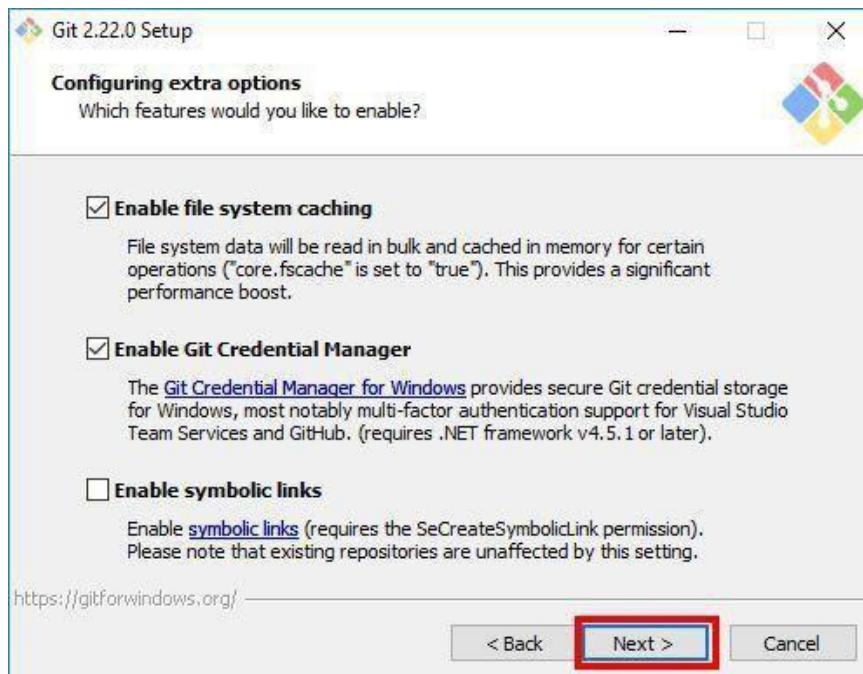
## Step 11: Configure the Terminal Emulator to use with Git BshPermalink

Next, select the terminal emulator you want Git Bash to use. I will proceed with the default option "Use MinTTY(the default terminal of MSYS2) and click "Next".



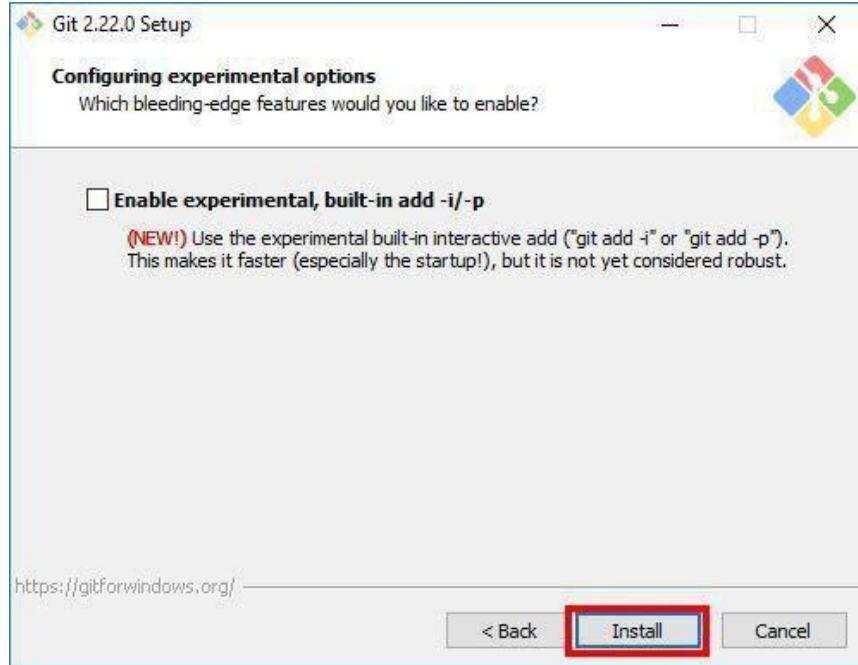
## Step 12: Configuring Extra Options

Select the features you want (the default options are fine) and click “Next”.



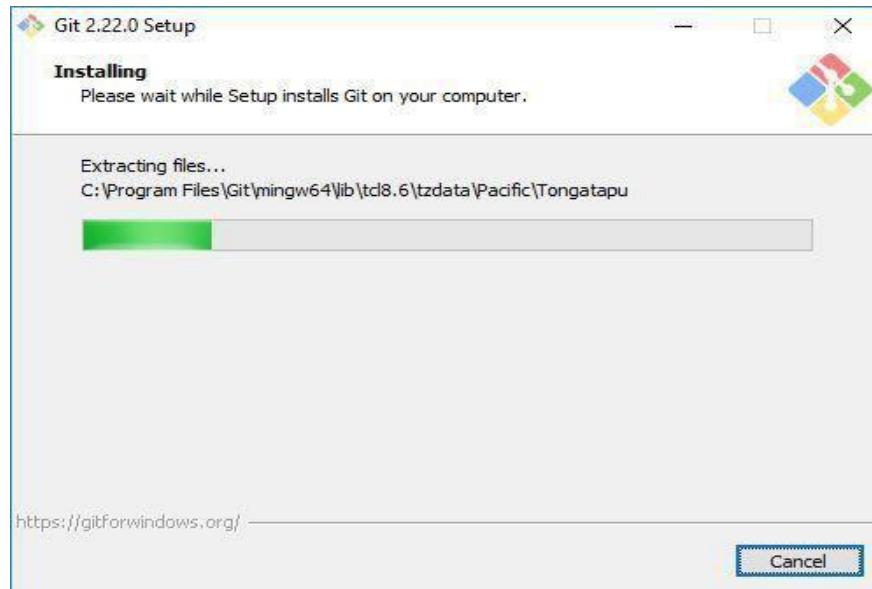
## Step 13: Configuring Extra Options

Enable experimental options if you want. Enabling them allows you to try out newer features that are still in development. I don't enable this, so I will just proceed by clicking “Install” to start the installation process.



## Step 14: Wait for Installation

Now, wait for a few minutes as the Setup Wizard installs Git on your computer.



## Step 15: Complete the Git Setup Wizard

After the installation has finished, check the “Launch Git Bash” and click “Finish” launching Git Bash.

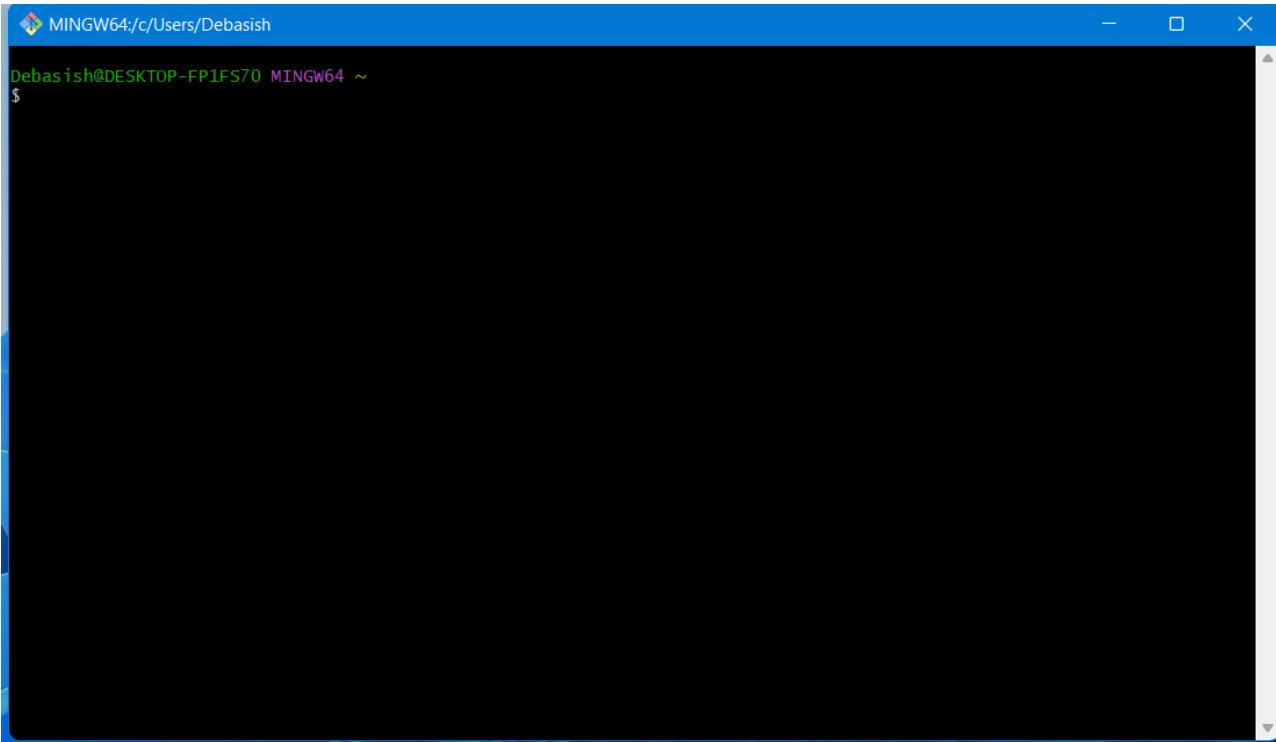


## Git –version

Run this command in command prompt to know the git version and git is successfully installed or not.

A screenshot of a Windows Command Prompt window titled "Command Prompt". It shows the command "git --version" being run and the output "git version 2.37.2.windows.2". The prompt then changes to "(venv) C:\Users\Debasish\Desktop\Major Project\Project>".

The Git Bash terminal will now open and you will be able to enter Git and Bash commands.

A screenshot of a Windows-style terminal window titled "MINGW64:/c/Users/Debasish". The title bar is blue with white text. The main area of the window is black, indicating it is currently empty or has no output to display.

Congratulations on successfully installing Git Bash.

# CHAPTER-5

---

## Selected Software or Platform

---

### What is Visual Studio Code?

**Visual Studio Code** (famously known as **VS Code**) is a free open-source text editor by Microsoft. VS Code is available for Windows, Linux, and macOS. Although the editor is relatively lightweight, it includes some powerful features that have made VS Code one of the most popular development environment tools in recent times.

### Why we use Visual Studio Code.

#### Edit, build, and debug with ease

At its heart, Visual Studio Code features a lightning-fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease.

#### Make it your own

Customize every feature to your liking and install any number of third-party extensions. While most scenarios work "out of the box" with no configuration, VS Code also grows with you, and we encourage you to optimize your experience to suit your unique needs. VS Code is an open-source project so you can also contribute to the growing and vibrant community on GitHub.

#### Built with love for the Web

VS Code includes enriched built-in support for Node.js development with JavaScript and TypeScript, powered by the same underlying technologies that drive Visual Studio. VS Code also includes great tooling for web technologies such as JSX/React, HTML, CSS, SCSS, Less, and JSON.

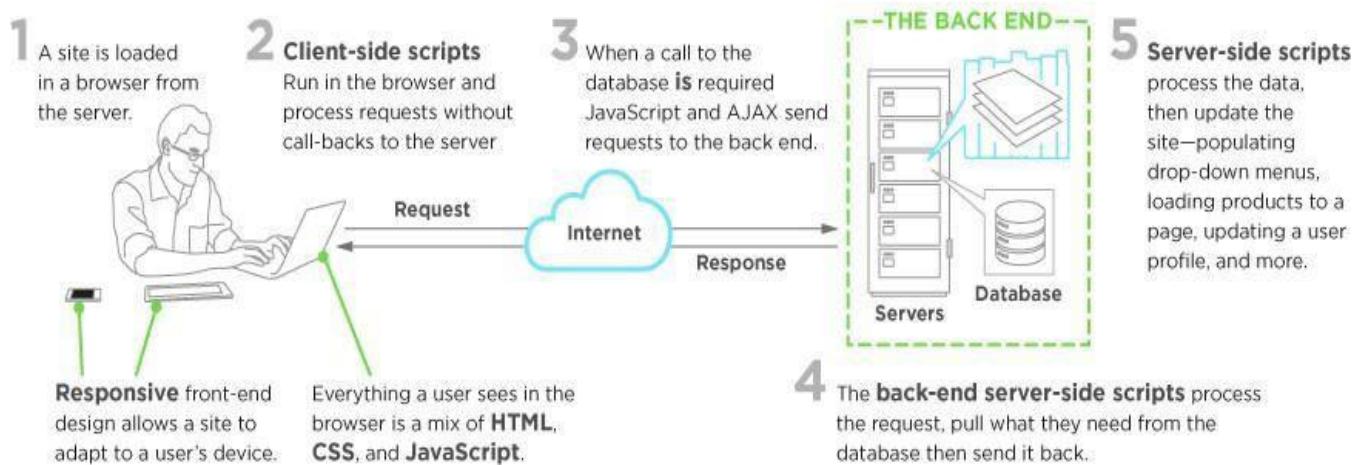
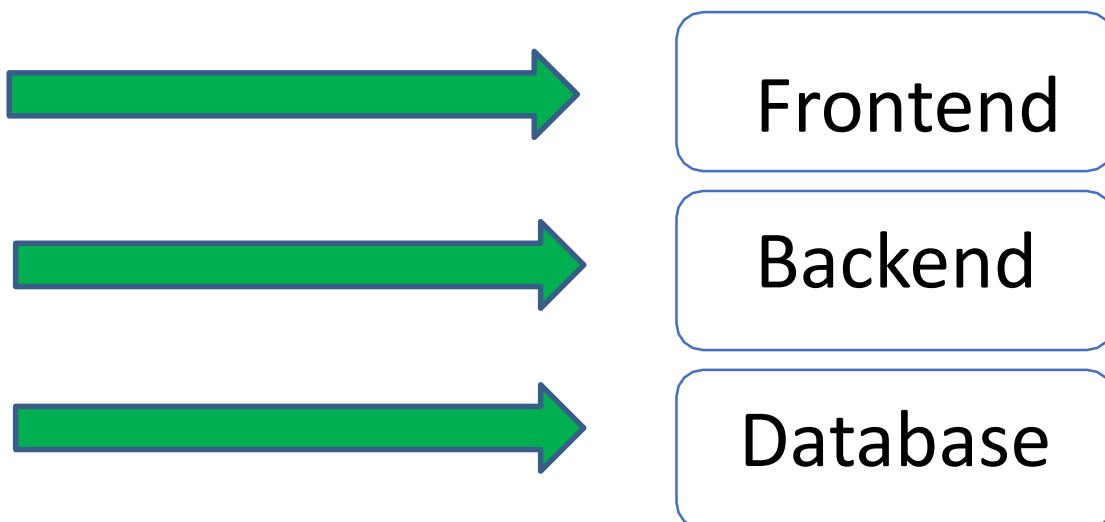
#### Robust and extensible architecture

Architecturally, Visual Studio Code combines the best of web, native, and language-specific technologies. Using Electron VS Code combines web technologies such as JavaScript and Node.js with the speed and flexibility of native apps. VS Code uses a newer, faster version of the same industrial-strength HTML-based editor that has powered the "Monaco" cloud editor, Internet Explorer's F12 Tools, and other projects. Additionally, VS Code uses a tools service architecture that enables it to integrate with many of the same technologies that power Visual Studio, including Roslyn for .NET, TypeScript, the Visual Studio debugging engine, and more.

Visual Studio Code includes a public extensibility model that lets developers build and use extensions, and richly customize their edit-build-debug experience.

## How to Developed a Project.

In every Application Development or Project Development basically Follow Three basic step: -



## Frontend Development

### What is Front end Development ?

Front-end development is the process of building the **visual** and **interactive** parts of a website or web application—the parts that users see and interact with. It focuses on creating a smooth and engaging user experience.

The three main languages you need to know well are HTML, CSS, and JavaScript. From there you can focus on frameworks, libraries, and other useful tools.

## HTML

HTML stands for Hypertext Markup Language. HTML displays the content on the page like buttons, links, headings, paragraphs, and lists.

## CSS

CSS stands for Cascading Style Sheets. CSS is responsible for the style of your web page including colors, layouts, and animations.

## JavaScript

JavaScript allows users to interact with the web page. Examples of JavaScript can be found in virtually any web page including the rimedu.com homepage.

Frameworks like [Bootstrap](#) allow you to add the catalogue of classes to your webpage. As a result, you end up with professional and mobile-friendly designs.

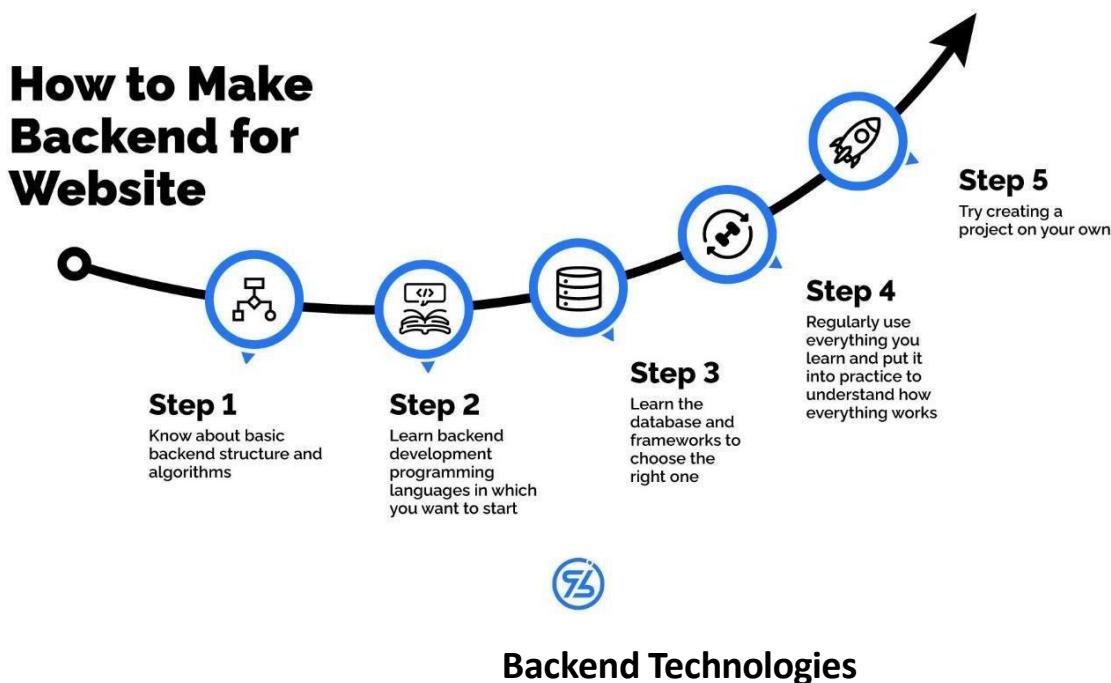
There are dozens of options on the market and you don't need to learn them all. It's often helpful

to look at jobs in your area and see what technologies they're using. Then you can focus on the most common/in-demand skills.

## What is Backend Development?

Back-end development means working on server-side software, which focuses on everything you can't see on a website. Back-end developers ensure the website performs correctly, focusing on databases, back-end logic, application programming interface (APIs), architecture, and servers. They use code that helps browsers communicate with databases, store, understand, and delete data.

On a team, back-end developers collaborate with front-end developers, product managers, principal architects, and website testers to build the structure of a website or mobile app. Back-end developers must be familiar with many kinds of tools and frameworks, including languages such as Python, Java, and Ruby. They make sure the back-end performs quickly and responsively to front-end user requests.



## Database

A **database** is an organized collection of data, so that it can be easily accessed and managed.

A database is a systematic collection of data. They support electronic storage and manipulation of data. Databases make data management easy.

Let us discuss a database example: An online telephone directory uses a database to store data of people, phone numbers, and other contact details. Your electricity service provider uses a database to manage billing, client-related issues, handle fault data, etc.

### How to Design a Database?

It is important for a Software Development Professional to know how to design the Database. The main objective of the database design process is to build a robust database.

The database Design is a crucial step in the Software development life cycle (SDLC) that involves Database Operations. The database operations including storing, retrieving and manipulating the data.

The Database Design a specialist job that requires some special Skills in terms of planning the database and then actually building the database using a specific database management system.



# Installation process of Mysql Workbench in windows.

Step 1: Download MySQL Workbench

Go to the official MySQL Workbench download page.

Select your operating system (Microsoft Windows).

Click on "Download" under the latest version.

If prompted, you can either log in or click "No thanks, just start my download."

The screenshot shows a web browser window with the URL [dev.mysql.com/downloads/workbench/](https://dev.mysql.com/downloads/workbench/). The page title is "MySQL Community Downloads". Under "General Availability (GA) Releases", the "MySQL Workbench 8.0.41" section is highlighted. A dropdown menu "Select Operating System" is set to "Microsoft Windows". The "Recommended Download" is the "MySQL Installer for Windows", which is described as "All MySQL Products. For All Windows Platforms. In One Package." It features a logo with a dolphin and four squares. Below it, there's a note about the MySQL Installer replacing standalone MSI packages. A link "Windows (x86, 32 & 64-bit), MySQL Installer MSI" leads to the download page. Other download options include "Windows (x86, 64-bit), MSI Installer" and "Windows (x86, 32-bit), ZIP Archive". A note at the bottom suggests using MD5 checksums and GnuPG signatures for package integrity verification.

## Step 2: Install MySQL Workbench

Open the downloaded .msi installer.

Click Next to begin installation.

Select "Complete" installation and click Next.

Click Install and wait for the installation to complete.

Click Finish once the installation is done.



## Step 3: Open MySQL Workbench

Search for MySQL Workbench in the Windows Start menu and open it.

## To setup User Username And Password

Step 1: Open MySQL Workbench

Launch MySQL Workbench from the Start menu (Windows) or Applications (Mac/Linux).

Step 2: Connect to MySQL Server

Click on "Local instance MySQL" (or create a new connection).

If prompted for a password, leave it blank (if not set yet) and click OK.

If the connection fails, you may need to start MySQL Server:

Windows: Open Services, find MySQL80 (or similar), and start it.

Mac/Linux: Run:

**sudo systemctl start mysql**

Step 3: Set a Root Password

Method 1: Using MySQL Workbench

In MySQL Workbench, go to Server > Users and Privileges.

Select root under Users.

Click Change Password, enter a new password, and confirm.

Click Apply.

Method 2: Using SQL Command

Open a new SQL query tab.

Run this command:

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'YourNewPassword';
FLUSH PRIVILEGES;
```

Click Execute (the lightning icon).

Step 4: Allow Root Access from Anywhere (Optional)

By default, MySQL only allows root access from the local machine. To enable remote access:

```
ALTER USER 'root'@'%' IDENTIFIED BY 'YourNewPassword';
FLUSH PRIVILEGES;
```

Note: Enabling remote root access can be a security risk. Use it only if necessary.

Step 5: Restart MySQL Server

After setting the password, restart the MySQL server:

Windows: Restart the MySQL Service.

Mac/Linux:

```
sudo systemctl restart mysql
```

# Sql Queries:-

The screenshot shows a SQL editor interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and navigation arrows. A search bar on the right contains the text "fash-in". The left sidebar features various icons for file operations like copy, paste, search, and refresh, with a count of 6 notifications.

The main area displays a script named "script.sql" with the following content:

```
CREATE SCHEMA fash_in;
use fash_in;

CREATE TABLE user (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    user_name VARCHAR(255) NOT NULL,
    user_email VARCHAR(255) UNIQUE NOT NULL,
    user_password VARCHAR(255) NOT NULL,
    user_phone VARCHAR(20),
    user_address TEXT,
    user_role ENUM('USER','ADMIN') DEFAULT 'USER',
    created_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP ,
    created_by VARCHAR(255),
    updated_by VARCHAR(255)
);

CREATE TABLE category (
    category_id VARCHAR(255) PRIMARY KEY ,
    category_title VARCHAR(255) NOT NULL,
    category_description TEXT NULL,
    category_price DECIMAL(10,2) NOT NULL,
    category_image_url TEXT NOT NULL,
    category_status ENUM ('ACTIVE','INACTIVE') DEFAULT 'ACTIVE'
    created_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    created_by VARCHAR(255),
    updated_by VARCHAR(255)
);

CREATE TABLE order_table (
    order_id VARCHAR PRIMARY KEY,
    user_id INT NOT NULL,
    total_amount DECIMAL(10,2) NOT NULL,
    order_status ENUM('PENDING', 'SHIPPED', 'DELIVERED', 'CANCELLED','CART') DEFAULT 'PENDING',
    created_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    created_by VARCHAR(255),
    updated_by VARCHAR(255),
    FOREIGN KEY (user_id) REFERENCES user(id) ON DELETE CASCADE
);

CREATE TABLE order_list (
    order_id VARCHAR PRIMARY KEY,
    order_item_id VARCHAR(255) PRIMARY KEY,
```

At the bottom, a status bar indicates "main\* ② ⑧ 0 ▲ 12 Indexing completed." and "Java: Ready".



File Edit Selection View Go Run Terminal Help ← → ⌘ fash-in

 *script.sql* 

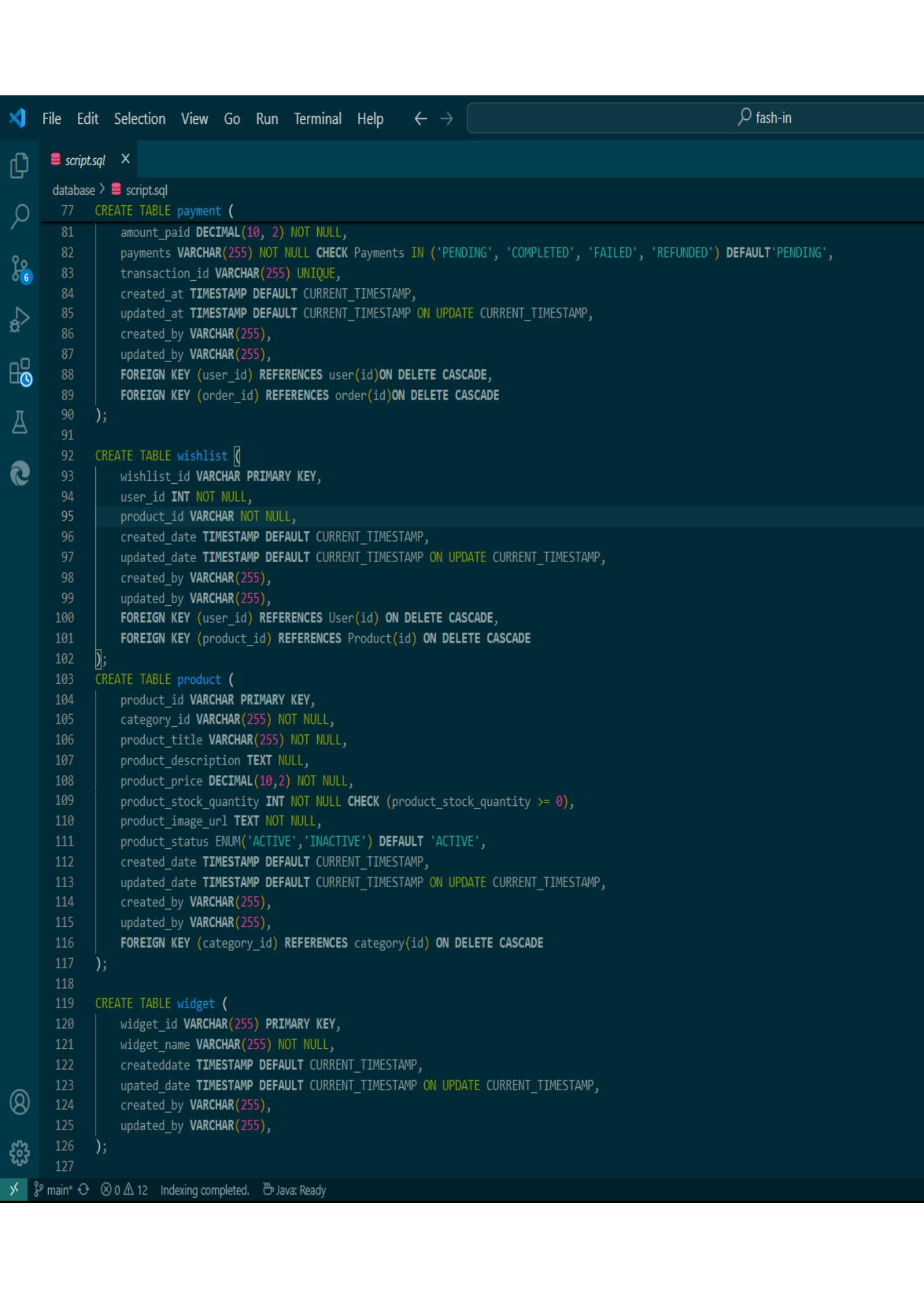
database >  script.sql

```
43 CREATE TABLE order_table (
44     FOREIGN KEY (user_id) REFERENCES user(id) ON DELETE CASCADE
45 );
46
47 CREATE TABLE order_list (
48     order_id VARCHAR PRIMARY KEY,
49     order_item_id VARCHAR(255) PRIMARY KEY,
50     user_id INT NOT NULL,
51     order_date DATE NOT NULL,
52     product_id VARCHAR(255) NOT NULL,
53     quantity INT NOT NULL,
54     order_item_unit_price DECIMAL(10,2) NOT NULL,
55     total_amount DECIMAL(10, 2) NOT NULL,
56     created_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
57     updated_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
58     created_by VARCHAR(255),
59     updated_by VARCHAR(255),
60     FOREIGN KEY (order_id) REFERENCES order(id)ON DELETE CASCADE,
61     FOREIGN KEY (product_id) REFERENCES Product(id)ON DELETE CASCADE
62 );
63
64 CREATE TABLE review (
65     review_id VARCHAR(255) PRIMARY KEY,
66     product_id VARCHAR(255) NOT NULL,
67     user_id INT NOT NULL,
68     rating INT NOT NULL CHECK (Rating >= 1 AND Rating <= 5),
69     review_text TEXT,
70     created_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
71     updated_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
72     created_by VARCHAR(255),
73     updated_by VARCHAR(255),
74     FOREIGN KEY (product_id) REFERENCES Product(id)ON DELETE CASCADE,
75     FOREIGN KEY (user_id) REFERENCES user(id)ON DELETE CASCADE
76 );
77
78 CREATE TABLE payment (
79     payment_id VARCHAR(255) PRIMARY KEY,
80     order_id VARCHAR NOT NULL,
81     payment_method VARCHAR(50) NOT NULL,
82     amount_paid DECIMAL(10, 2) NOT NULL,
83     payments VARCHAR(255) NOT NULL CHECK Payments IN ('PENDING', 'COMPLETED', 'FAILED', 'REFUNDED') DEFAULT 'PENDING',
84     transaction_id VARCHAR(255) UNIQUE,
85     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
86     updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
87     created_by VARCHAR(255),
88     updated_by VARCHAR(255),
89     FOREIGN KEY (user_id) REFERENCES user(id)ON DELETE CASCADE,
90     FOREIGN KEY (order_id) REFERENCES order(id)ON DELETE CASCADE
```

✖ ⚡ main\* ⓘ ⓘ 0 △ 12 Indexing completed. 🎵 Java: Ready

script.sql X

```
database > script.sql
63 CREATE TABLE review (
75 );
76
77 CREATE TABLE payment (
78     payment_id VARCHAR(255) PRIMARY KEY,
79     order_id VARCHAR NOT NULL,
80     payment_method VARCHAR(50) NOT NULL,
81     amount_paid DECIMAL(10, 2) NOT NULL,
82     payments VARCHAR(255) NOT NULL CHECK Payments IN ('PENDING', 'COMPLETED', 'FAILED', 'REFUNDED') DEFAULT 'PENDING',
83     transaction_id VARCHAR(255) UNIQUE,
84     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
85     updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
86     created_by VARCHAR(255),
87     updated_by VARCHAR(255),
88     FOREIGN KEY (user_id) REFERENCES user(id) ON DELETE CASCADE,
89     FOREIGN KEY (order_id) REFERENCES order(id) ON DELETE CASCADE
90 );
91
92 CREATE TABLE wishlist (
93     wishlist_id VARCHAR PRIMARY KEY,
94     user_id INT NOT NULL,
95     product_id VARCHAR NOT NULL,
96     created_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
97     updated_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
98     created_by VARCHAR(255),
99     updated_by VARCHAR(255),
100    FOREIGN KEY (user_id) REFERENCES User(id) ON DELETE CASCADE,
101    FOREIGN KEY (product_id) REFERENCES Product(id) ON DELETE CASCADE
102 );
103 CREATE TABLE product (
104     product_id VARCHAR PRIMARY KEY,
105     category_id VARCHAR(255) NOT NULL,
106     product_title VARCHAR(255) NOT NULL,
107     product_description TEXT NULL,
108     product_price DECIMAL(10,2) NOT NULL,
109     product_stock_quantity INT NOT NULL CHECK (product_stock_quantity >= 0),
110     product_image_url TEXT NOT NULL,
111     product_status ENUM('ACTIVE', 'INACTIVE') DEFAULT 'ACTIVE',
112     created_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
113     updated_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
114     created_by VARCHAR(255),
115     updated_by VARCHAR(255),
116     FOREIGN KEY (category_id) REFERENCES category(id) ON DELETE CASCADE
117 );
118
119 CREATE TABLE widget (
120     widget_id VARCHAR(255) PRIMARY KEY,
121     widget_name VARCHAR(255) NOT NULL,
```



# SnapShot of User Table

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator product widget product user product category category widget user user

SCHEMAS fash\_in

Tables category order\_item orders payment product ratings user widget widget\_product wishlist

Views

Stored Procedures

Functions

sakila

sys

world

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

1 • SELECT \* FROM fash\_in.user;

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

ed_date	created_by	created_date	updated_by	user_address	user_email	user_password	user_phone	user_role
13-26 20:18:26.112535	user	2025-03-26 20:18:26.112535	user	HULL	aditya@gmail.com	\$2a\$12\$hzS3.42/rNU5orp6OhhejDEpEM3HQs...	7008905821	ADMIN
13-27 18:51:58.649090	user	2025-03-27 18:51:58.649090	user	HULL	Simran@gmail.com	\$2a\$12\$8bckYzLCKUe64XhezRvR.i.2IB56xPYz...	6370585830	USER

Administration Schemas

Information

Table: user

Columns:

user_id	bigint A
updated_date	datetime
created_by	varchar(
created_date	datetim
updated_by	varchar(
user_address	text
user_email	varchar(
user_password	varchar(
user_phone	varchar(
user_role	enum('')
user_name	mediumn

Object Info Session

# SnapShot of Product Table:-

MySQL Workbench Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator product widget product user product

SCHEMAS fash\_in

Tables category order\_item orders payment product ratings user widget widget\_product wishlist

Views

Stored Procedures Functions

sakila sys world

Administration Schemas

Information

Table: product

Columns:

- product\_id
- updated\_date
- created\_by
- created\_date
- updated\_by
- product\_description
- product\_image\_url
- product\_price
- product\_status
- product\_stock\_quantity
- product\_type

Object Info Session

SQLAdditions SQL Editor Jump to

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or toggle automatic help.

Result Grid Filter Rows: Edit Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types

1 • SELECT \* FROM fash\_in.product;

product_id	updated_date	created_by	created_date	updated_by	product_description	product_image_url
02a65e9e-4ac3-4333-a29d-438bd9f2c50c	2025-02-22 13:49:28.000000	admin	2025-02-22 13:49:28.000000	admin	A gentle facial bleach that brightens the skin an...	https://ex...
15d7fc25-93b3-45e7-b33e-18a9d0f5c6f2	2025-02-22 13:49:28.000000	admin	2025-02-22 13:49:28.000000	admin	A luxurious silk scarf with a vibrant pattern, add...	https://ex...
1b663720-6580-4319-b41b-89cf19503999	2025-02-22 13:49:28.000000	admin	2025-02-22 13:49:28.000000	admin	Bright and fun rain boots for kids, keeping their ...	https://ex...
1b8464bc-23a5-4933-b3b5-3c2c727ceda5	2025-02-22 13:49:28.000000	admin	2025-02-22 13:49:28.000000	admin	Stylish cargo shorts with multiple pockets, perfe...	https://ex...
1c72d153-1fe7-4a1f-b0b0-2337128f6472	2025-02-22 13:49:28.000000	admin	2025-02-22 13:49:28.000000	admin	Premium tape-in hair extensions that seamlessly...	https://ex...
1f27c7d9-2630-4f5f-b6e6-b2c1a2d9c3c7	2025-02-22 13:49:28.000000	admin	2025-02-22 13:49:28.000000	admin	A gentle facial bleach that brightens the skin an...	upload_im...
249c35bf-44df-4f55-9206-9b316ce19059	2025-03-21 03:42:29.429679	NULL	2025-03-21 03:42:29.429679	NULL	A gentle facial bleach that brightens the skin an...	upload_im...
2d5a5cf-fd3f-426b-a1b4-52a0c3bb9f53	2025-02-22 13:49:28.000000	admin	2025-02-22 13:49:28.000000	admin	A facial roller designed to stimulate blood circula...	https://ex...
2db0ba36-4b3c-4f45-bae6-18340e239d97	2025-03-21 03:38:36.300558	NULL	2025-03-21 03:38:36.300558	NULL	student of jt	upload_im...

Output

Action Output

#	Time	Action
1	18:43:49	SELECT * FROM fash_in.product LIMIT 0, 1000

Message 106 row(s) returned Duration / Fetch 0.000 sec / 0.000 sec

# SnapShot of Category Table

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator product widget product user product category category ×

SCHEMAS fash\_in

Tables category order\_item orders payment product ratings user widget widget\_product wishlist

Views

Stored Procedures

Functions

sakila

sys

world

SQLAdditions... SQL Editor Jump to

Automatic context help is disabled. Use the toolbar or manually get help for the current caret position or toggle automatic help.

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types

category_id	updated_date	created_by	created_date	updated_by	category_description	category_image_url
a1234567-89ab-4cde-1234-56789abcdef0	2025-02-19 10:00:00.000000	admin	2025-02-19 10:00:00.000000	admin	A wide variety of stylish and comfortable clothing pieces designed for...	https://example.com/categories/1
b1234567-89ab-4cde-1234-56789abcdef1	2025-02-19 10:00:00.000000	admin	2025-02-19 10:00:00.000000	admin	Trendy and elegant clothing pieces designed for...	https://example.com/categories/2
c1234567-89ab-4cde-1234-56789abcdef2	2025-02-19 10:00:00.000000	admin	2025-02-19 10:00:00.000000	admin	Comfortable and playful clothing for children of...	https://example.com/categories/3
d1234567-89ab-4cde-1234-56789abcdef3	2025-02-19 10:00:00.000000	admin	2025-02-19 10:00:00.000000	admin	A selection of perfumes and colognes for all pre...	https://example.com/categories/4
e1234567-89ab-4cde-1234-56789abcdef4	2025-02-19 10:00:00.000000	admin	2025-02-19 10:00:00.000000	admin	High-quality skincare products for a radiant, he...	https://example.com/categories/5
f1234567-89ab-4cde-1234-56789abcdef5	2025-02-19 10:00:00.000000	admin	2025-02-19 10:00:00.000000	admin	Products to help maintain beautiful and healthy...	https://example.com/categories/6
g1234567-89ab-4cde-1234-56789abcdef6	2025-02-19 10:00:00.000000	admin	2025-02-19 10:00:00.000000	admin	Makeup and other beauty products to enhance ...	https://example.com/categories/7
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table: category

Columns:

category_id	VARCHAR
updated_date	DATE
created_by	VARCHAR
created_date	DATE
updated_by	VARCHAR
category_description	TEXT
category_image_url	TEXT
category_price	DECIMAL
category_status	BOOLEAN
category_title	VARCHAR

Action Output

#	Time	Action	Message	Duration / Fetch
1	18:45:04	SELECT * FROM fash_in.category LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

# Software Requirements Specification

**Purpose** FashIn is a fashion and lifestyle web application designed to provide users with a seamless shopping experience for trendy clothing, accessories, and fashion-related content. This document outlines the functional and non-functional requirements of the application, ensuring a clear development roadmap.

**Scope** FashIn allows users to browse and purchase fashion products and explore fashion trends. The application will include features like user registration, product search, cart management and order processing.

## Definitions, Acronyms, and Abbreviations

- **SRS** – Software Requirements Specification
- **UI** – User Interface
- **DBMS** – Database Management System
- **CRUD** – Create, Read, Update, Delete

## References

- Web Technologies: HTML, CSS, JavaScript, Java (Spring Boot)
- Database: MySQL

**Overview** This document covers functional and non-functional requirements, system features, constraints, and dependencies.

## 2. Overall Description

**Product Perspective** FashIn is an independent web application for fashion and lifestyle shopping. It follows an e-commerce model with an integrated fashion blog to engage users.

### 2. Product Functions

- User Registration and Authentication
- Product Category
- Shopping Cart and Checkout
- Order Management

### 3. User Classes and Characteristics

- **Customers** – Browse and purchase fashion items.
- **Admin** – Manage products, orders, and edit or Delete Items.

### 4. Operating Environment

- Frontend: HTML, CSS, JavaScript
- Backend: Spring Boot (Java)
- Database: MySQL

## 5. Constraints

- Responsive design for mobile and desktop.
- Secure user authentication.

## 3. Specific Requirements

### 1. Functional Requirements

- **User Management:**
  - Users can Sign-Up, Sign-In their profiles.
- **Product Management:**
  - Admin can add, edit, or delete products.
- **Cart and Order Processing:**
  - Users can add/remove products from the cart.

## 4. System Features

1. **Registration & Login** Users can create accounts and log in.
2. **Product Catalogue** Users can browse fashion products and view detailed descriptions.
3. **Shopping Cart & Checkout** Users can add products to the cart.
4. **Admin Dashboard** Admins can manage products and orders.

## 5. Other Requirements

- Responsive design for different screen sizes.

# CHAPTER-7

## SYSTEM DESIGN

### System Requirements Definition

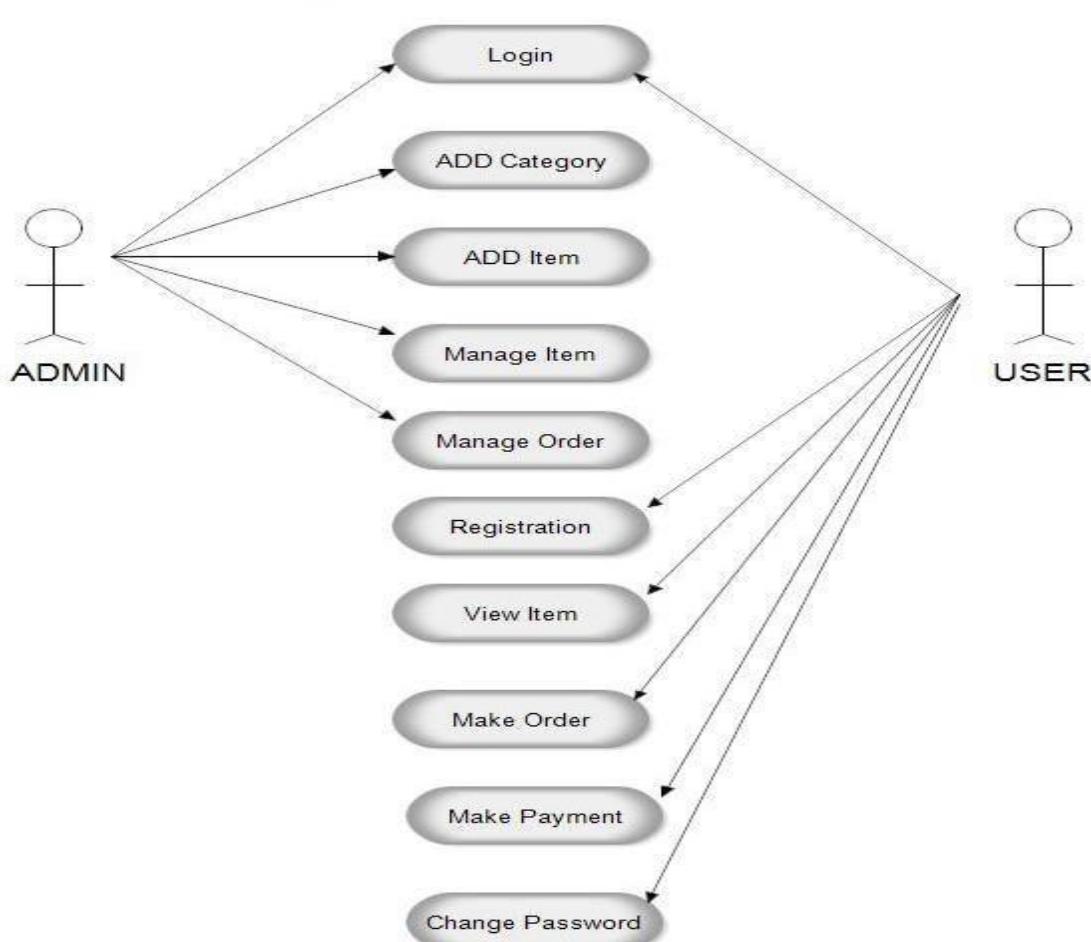
System requirement definitions specify what the system should do, its functionality and its essential and desirable system properties. The techniques applied to elicit and collect information in order to create system specifications and requirement definitions involve consultations, interviews, requirements workshop with customers and end users. The objective of the requirements definition phase is to derive the two types of requirement:

#### Functional requirements

They define the basic functions that the system must provide and focus on the need goals of the end users.

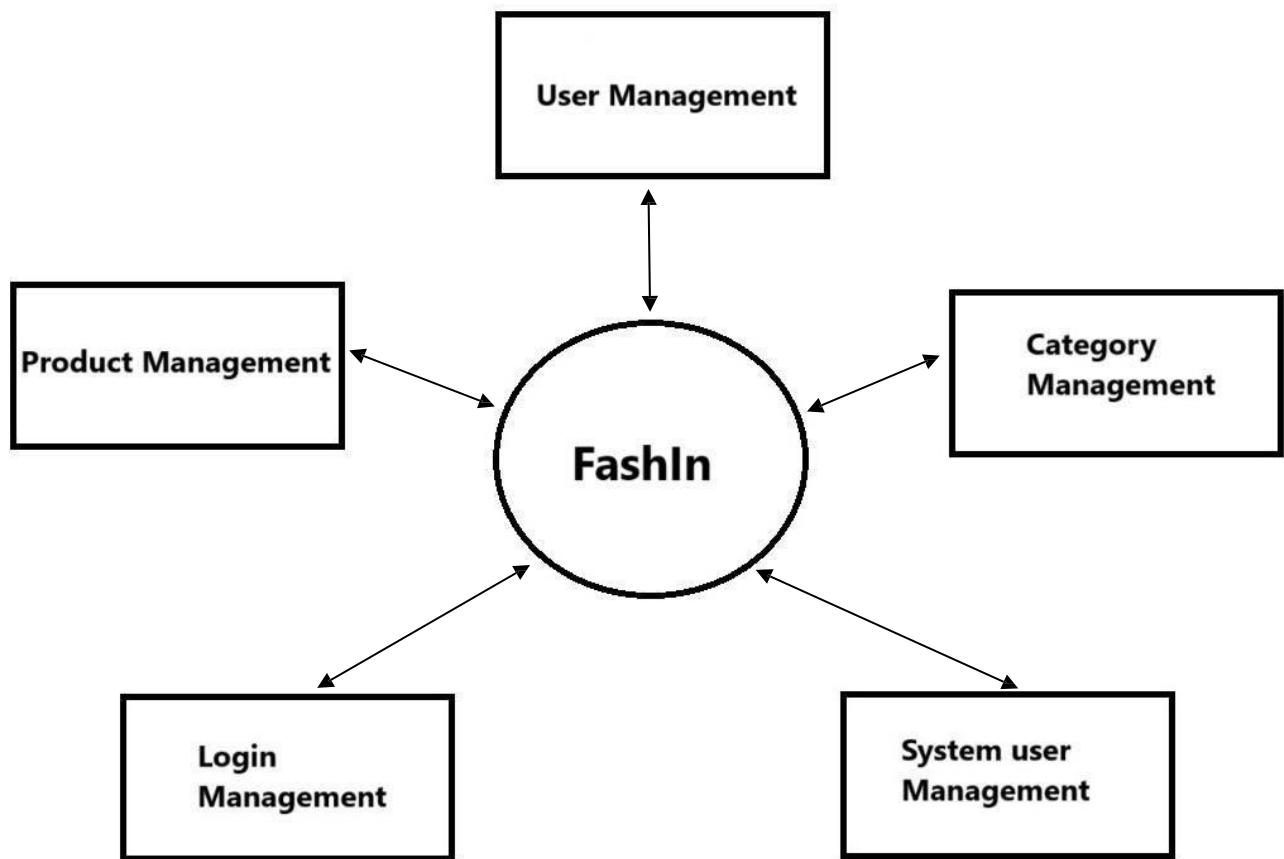
### Use-case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. In our system User will interact with use cases like Capture Image, Audio Input, Save text, Retrieve Text, Audio output.

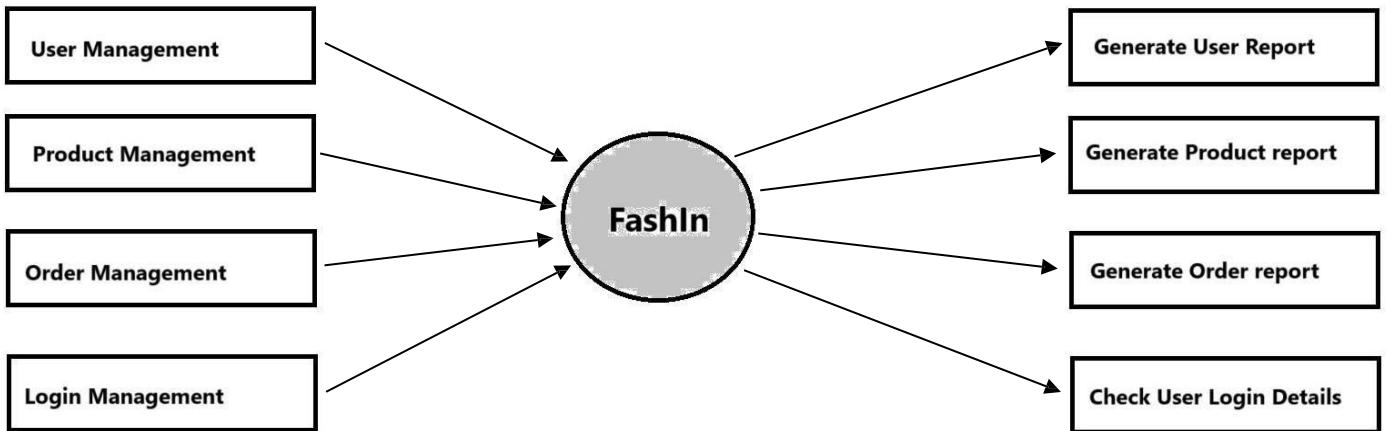


## Data-flow Diagram

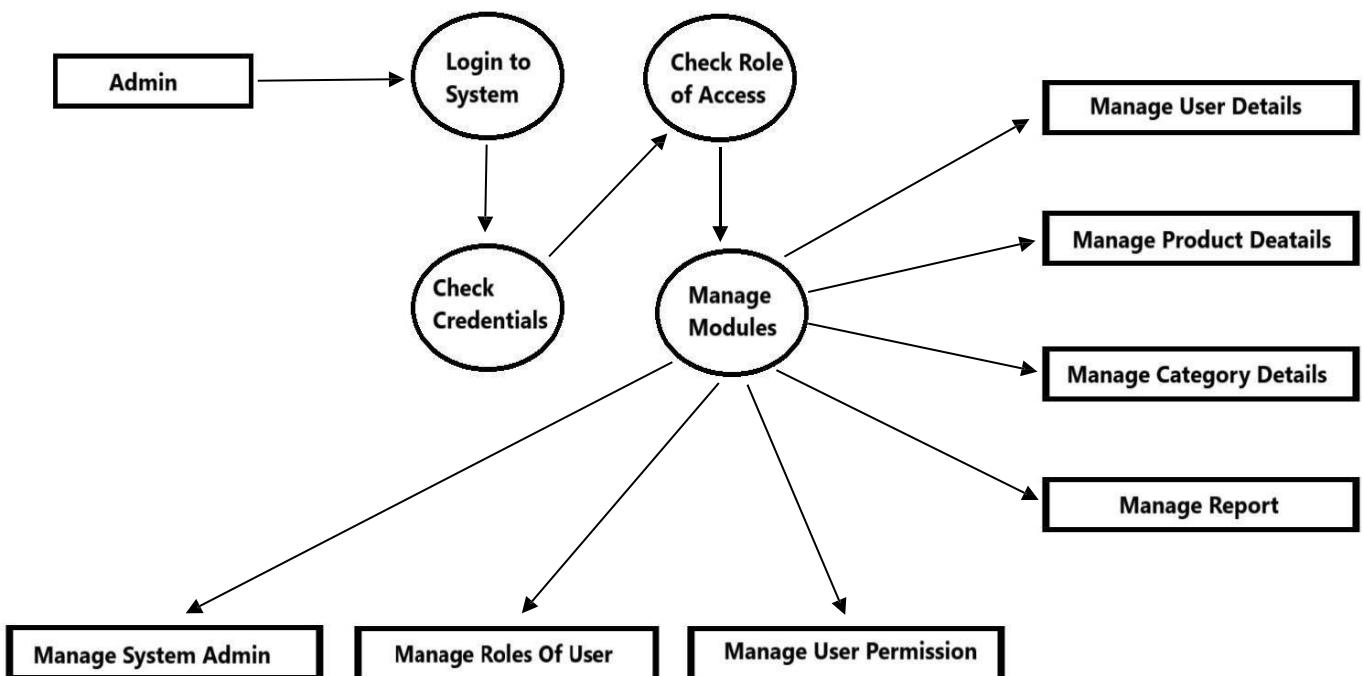
A data-flow diagram is a way of representing a flow of a data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. Given below is Level 0 Level 1 and Level 2 DFD of system.



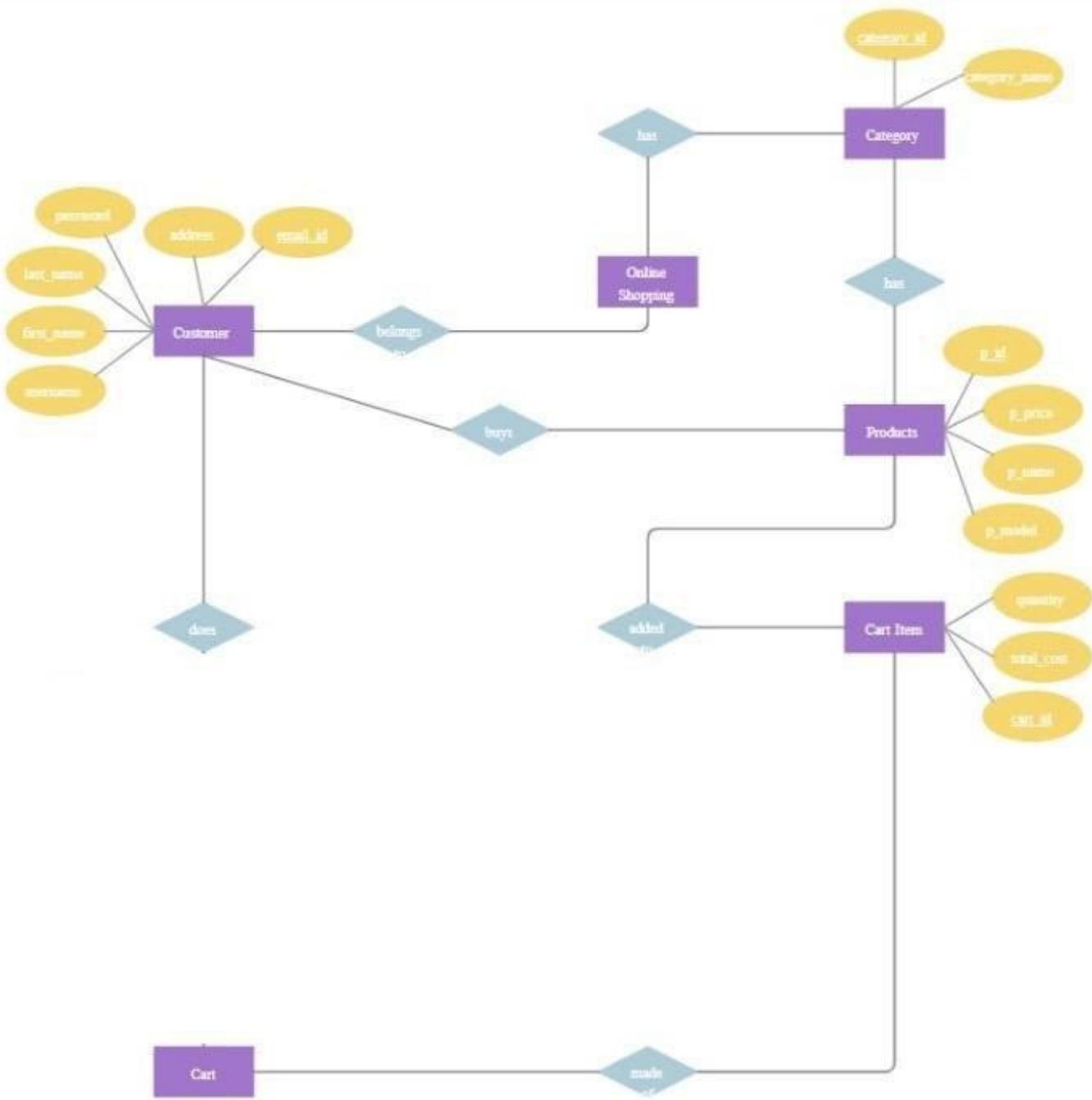
0 Level DFD Diagram



Level 1 DFD Diagram



Level 2 DFD Diagram



ER Diagram

## System requirements (non-functional requirements)

These are non-functional system properties such as availability, performance and safety etc. They define functions of a system, services and operational constraints in detail.

File Edit Selection View Go Run Terminal Help ← → ⌘ fash-in ⌘ v

home.html X

src > main > resources > templates > customer > home.html > html > body > div.static-image > a

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head data-th-replace="~{fragment/common :: head}"></head>
5
6  <body>
7    <link rel="stylesheet" href="/style/customer.home.css">
8
9
10   <nav data-th-replace="~{fragment/common :: nav}"></nav>
11   <div data-th-replace="~{fragment/common :: #searchdiv}"></div>
12   <div class="carousel-container">
13     <div class="carousel">
14       
15     </div>
16     <div class="carousel">
17       
20     <div class="carousel">
21       
24     <div class="carousel">
25       
26     </div>
27   </div>
28 </div>
29 <h2 class="product-category">Best Selling</h2>
30 <div class="product-elements">
31   <section class="product">
32     <button class="pre-btn">
33       
34     </button>
35     <div class="product-container">
36       <div class="product-card">
37         <div class="product-image">
38           <span class="discount-tag">50% off</span>
39           
42           <button class="card-btn">Add to Wishlist</button>
43         </div>
44         <div class="product-info">
45           <h2 class="product-brand">Brand</h2>
46           <p class="product-short-description">A short line about the product</p>
47           <span class="price">$20</span>
48           <span class="actual-price">$40</span>
49         </div>
50
51       <div class="product-card">
52         <div class="product-image">
53           <span class="discount-tag">50% off</span>
54           
57           <button class="card-btn">Add to Wishlist</button>
58         </div>
59         <div class="product-info">
60           <h2 class="product-brand">Brand</h2>
61           <p class="product-short-description">A short line about the product</p>
62           <span class="price">$20</span>
63           <span class="actual-price">$40</span>
64         </div>
65
66       <div class="product-card">
67         <div class="product-image">
68           <span class="discount-tag">50% off</span>
69           
72           <button class="card-btn">Add to Wishlist</button>
```

Indexing completed. ⌘ Java: Ready ⌘ AdityaKhandwal (1 month ago) Ln 279, Col 21 Spaces: 4 UTF-8 CRLF {} HTML ⌘ Go Live

File Edit Selection View Go Run Terminal Help ← → ⌘ fash-in ⌘ v

home.html X

src > main > resources > templates > customer > home.html > html > body > div.static-image > a

```
2  <html lang="en">
3  <body>
30   <div class="product-elements">
31     <section class="product">
35       <div class="product-container">
38         <span class="discount-tag">50% off</span>
39         
42         <button class="card-btn">Add to Wishlist</button>
43       </div>
44       <div class="product-info">
45         <h2 class="product-brand">Brand</h2>
46         <p class="product-short-description">A short line about the product</p>
47         <span class="price">$20</span>
48         <span class="actual-price">$40</span>
49       </div>
50
51     <div class="product-card">
52       <div class="product-image">
53         <span class="discount-tag">50% off</span>
54         
57         <button class="card-btn">Add to Wishlist</button>
58       </div>
59       <div class="product-info">
60         <h2 class="product-brand">Brand</h2>
61         <p class="product-short-description">A short line about the product</p>
62         <span class="price">$20</span>
63         <span class="actual-price">$40</span>
64       </div>
65
66     <div class="product-card">
67       <div class="product-image">
68         <span class="discount-tag">50% off</span>
69         
72         <button class="card-btn">Add to Wishlist</button>
```

Indexing completed. ⌘ Java: Ready ⌘ AdityaKhandwal (1 month ago) Ln 279, Col 21 Spaces: 4 UTF-8 CRLF {} HTML ⌘ Go Live

File Edit Selection View Go Run Terminal Help ← → ⌘ fash-in ⌘ v

home.html X

```
src > main > resources > templates > customer > home.html > html > body > div.static-image > a
  2   <html lang="en">
  6   <body>
 30     <div class="product-elements">
 31       <section class="product">
 35         <div class="product-container">
122           <span class="actual-price">$40</span>
123         </div>
124       </div>
126
127       <div>
128         <button class="next-btn">
129           
130         </button>
131       </div>
132     </div>
133   <div class="brands">
134     <h2 class="brands-heading">Hot Brands on Offers</h2>
135     <div class="product-container-1">
136       <div class="card-1">
137         <div class="product-1">
138           <div class="image">
139             <span class="offers">offers 20%</span>
140             <a href="#"></a>
143             <button class="brand-btn">Visit >></button>
144             <h2>Shop now</h2>
145             <span>Style your ways about the product</span>
146           </div>
147         </div>
148       </div>
149
150       <div class="card-1">
151         <div class="product-1">
152           <div class="image">
153             <span class="offers">offers 20% </span>
154             <a href="#">
157       </div>
```

File Edit Selection View Go Run Terminal Help ← → ⌘ fash-in ⌘ v

home.html X

```
src > main > resources > templates > customer > home.html > html > body > div.static-image > a
  2   <html lang="en">
  6   <body>
277     <div class="static-image">
278       <a href="#">
279         
280       </a>
281       <span></span>
282     </div>
283   <div class="container-large-cards">
284     <div class="wrapper-large-cards">
285       <div class="cards-large">
286         <div class="images-large">
287           
288           <span class="blur">Makeup</span>
289         </div>
290         <div class="info">
291           <span class="infos">Upto 40% Off</span>
292           <span class="infos-1">on entire brand</span>
293         </div>
294       </div>
295     </div>
296
297     <div class="cards-large">
298       <div class="images-large">
299         
300         <span class="blur">fragrance</span>
301       </div>
302       <div class="info">
303         <span class="infos">Upto 40% Off</span>
304         <span class="infos-1">on entire brand</span>
305       </div>
306     </div>
307
308     <div class="cards-large">
309       <div class="images-large">
310         
311         <span class="blur">Hair Care</span>
312       </div>
313       <div class="info">
```

File Edit Selection View Go Run Terminal Help ← → ⌘ fash-in ⌘ v

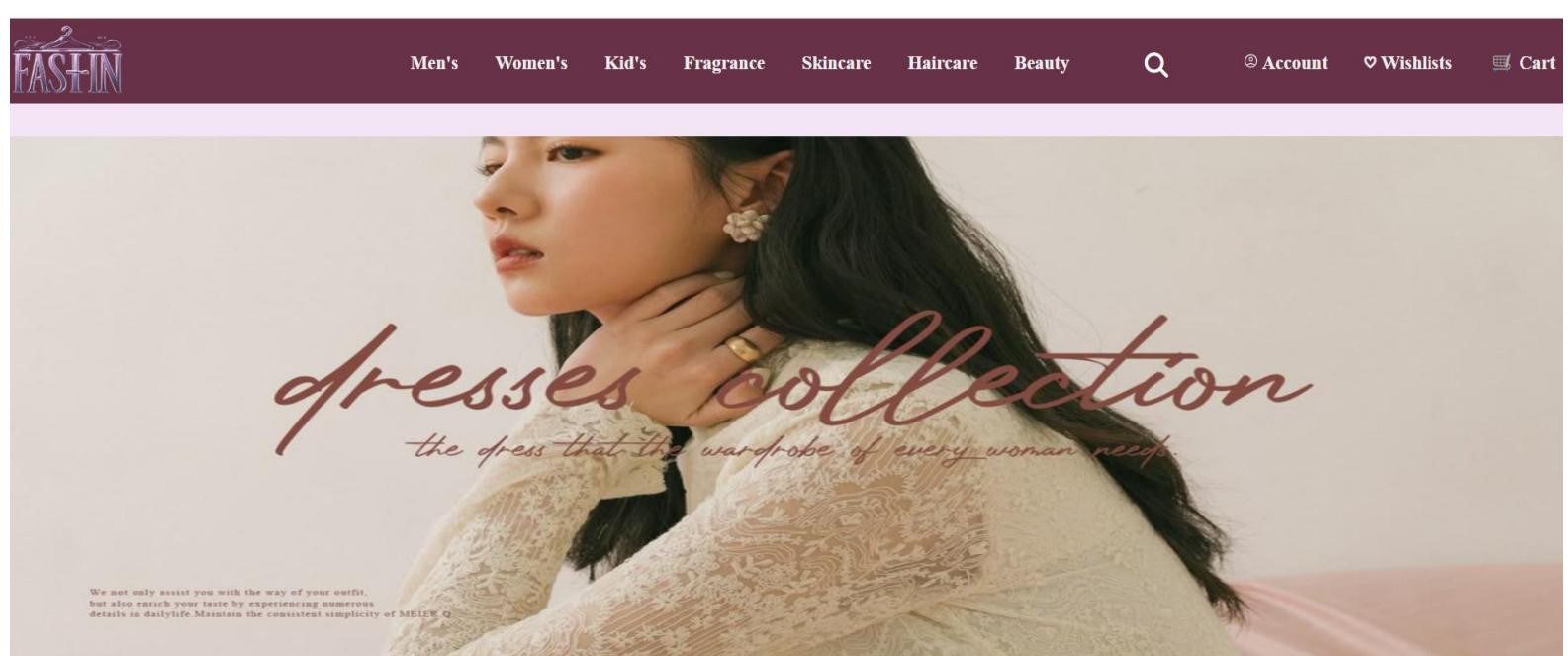
home.html X

src > main > resources > templates > customer > home.html > html > body > div.static-image > a

```
2   <html lang="en">
6   <body>
284     <div class="container-large-cards">
285       <div class="wrapper-large-cards">
318         |   </div>
319       </div>
320
321     <footer data-th-replace="~{fragment/common :: footer}"></footer>
322   <script>
323
324     function HomePage() {
325       const nextBtn = document.querySelector('.next-btn');
326       const prevBtn = document.querySelector('.pre-btn');
327       const productContainer = document.querySelector('.product-container');
328
329       nextBtn.addEventListener('click', () => {
330         |   productContainer.scrollBy({ left: 300, behavior: 'smooth' });
331       });
332       prevBtn.addEventListener('click', () => {
333         |   productContainer.scrollBy({ left: -300, behavior: 'smooth' });
334       });
335     }
336     HomePage();
337
338     function searchswipe() {
339       let button = document.getElementById("parentbtn");
340       let searchDiv = document.getElementById("searchdiv");
341       button.addEventListener("click", () => {
342         |   searchDiv.style.display = "block";
343       });
344     }
345     searchswipe();
346
347   </script>
348
349 </body>
350
351 </html>
```

✖ main\* ⓘ ⓘ 0 △ 12 Indexing completed. ⚡ Java: Ready ⌘ AdityaKhandwal (1 month ago) Ln 279, Col 21 Spaces: 4 UTF-8 CRLF {} HTML ⌘ Go Live

# SnapShot of HomePage



## Best Selling



## Best Selling



### BRAND

A Short Line About The Product

\$20 \$40



### BRAND

A Short Line About The Product

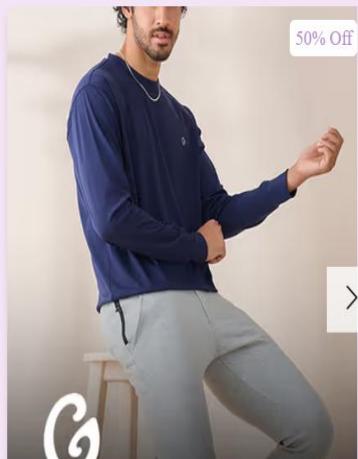
\$20 \$40



### BRAND

A Short Line About The Product

\$20 \$40

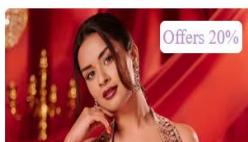


### BRAND

A Short Line About The Product

\$20 \$40

## Hot Brands On Offers



### Shop now

Style your ways about the product



### Shop now

Style your ways about the product



### Shop now

Style your ways about the product



### Shop now

Style your ways about the product



### Shop now

Style your ways about the product



### The Souled





## Makeup

Upto 40% Off  
on entire brand



## Fragrance

Upto 40% Off  
on entire brand



## Hair Care

Upto 40% Off  
on entire brand

Get Connected with us:

For any Help you may call us at  
7008905821 (Monday to Friday:10am - 9pm, Sunday:10am - 1pm)

[About us](#)  
[Careers](#)

[Help](#)  
[Return Policy](#)  
[Help Center](#)  
[Terms and Condition](#)

[Policy](#)  
[Sells on Fash-in](#)

[Quick Links](#)  
[offers](#)  
[sitemap](#)  
[collections](#)

**Women**  
Women Indian Wear  
Women Western wear  
Bags  
Women Footwear  
Women Sport's wear  
Women watches  
Fashion Accessories  
Bags  
Makeups

**Men**  
Topwear  
Bottomwear  
Ethnicwear  
Mens Footwear  
Mens Accessories  
Innerwear and sleepwear  
Men's Watches  
Bags and Bagpack  
Sports and fitness

**Kids**  
kid Indianwear  
Kids Westernwear  
Kids Footwear  
Kids Sleepwear  
Some Comfortwears  
Kids Sportswear  
Kids Accessories  
Kids Innerwear  
All Weather

**Top Brands**  
Forever New  
Cider  
Skechers  
Levis  
Puma  
Adidas  
Fablestreet  
Vero Moda  
Louis Vuitton

**Skin Care**  
Lip Care  
Eye Care  
Face Tool  
Moisturizer  
Facial Kits and Bleaches  
Toner and face Mists  
Masks  
Under eye Serums  
Essential Oil

**Hair Care**  
Complete Hair Accessories Collection  
Hair Color  
Hair Clips and Bands  
Hair Extensions  
Complete hair styling collection  
Toner and face Mists  
Toner and face Mists  
Complete Shampoo and Conditioner  
Styling creams gels and waxes  
Hair and Scalp Treatment oils

# Admin DashBoard

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="/style/admin/admin.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.1/css/all.min.css">
</head>
<body>
    <div class="admin-container">
        <div data-th-replace="~{'fragment/common' :: sidebar}"></div>

        <main class="main-content">
            <h1>Dashboard</h1>
            <div data-th-replace="~{'fragment/common' :: admin-cards}"></div>
            <h2>Manage Products</h2>
            <table>
                <thead>
                    <tr>
                        <th>ID</th>
                        <th>Product</th>
                        <th>Price</th>
                        <th>Stock</th>
                        <th>Action</th>
                    </tr>
                </thead>
                <tbody>
                    <tr data-th-each="product:${products}">
                        <td data-th-text="${product.productId}"></td>
                        <td data-th-text="${product.productTitle}"></td>
                        <td class="td" data-th-text="${product.productPrice}>$20</td>
                        <td class="td" data-th-text="${product.productStockQuantity}>65</td>
                        <td class="td">
                            <button class="button" id="editbutton1">Edit</button>
                            <button class="button">Delete</button>
                        </td>
                    </tr>
                </tbody>
            </table>
        </main>
    </div>

    <form id="form">
        <div id="container">
            <div class="addCategory">
```

```
File Edit Selection View Go Run Terminal Help ← → ⌂ fash-in

admin.html •

src/main/resources/templates/admin/admin.html
1 <html lang="en">
2   <body>
3     <div class="admin-container">
4       <main class="main-content">
5         </main>
6       </div>
7     </body>
8   </html>

9 <form id="form">
10   <div id="container">
11     <div class="addCategory">
12       <label for="addCategory">Add Category:</label>
13       <input class="inputbox" type="text" placeholder="Enter category">
14     </div>
15
16     <div class="addProduct">
17       <label for="addProduct">Add Product:</label>
18       <input class="inputbox" type="text" placeholder="Add your Product">
19     </div>
20
21     <div class="ProductImage">
22       <label for="ProductImage">Product Image:</label>
23       <input class="inputbox" type="file" placeholder="Product Image">
24     </div>
25
26     <div class="ProductDescription">
27       <label for="ProductDescription">Product Description:</label>
28       <input class="inputbox" type="text" placeholder="Product Description">
29     </div>
30
31     <div>
32       <button id="button1">Submit</button>
33     </div>
34   </div>
35 </form>
36
37 </body>
38
39 </html>
```

Indexing completed. Java Ready

# SnapShot of Admin Dashboard

The screenshot shows a web browser window with the URL [localhost:1211/admin/dashboard](http://localhost:1211/admin/dashboard). The page has a dark sidebar on the left and a light-colored main content area.

**Left Sidebar (Fashin Admin):**

- Dashboard
- Orders
- Products
- Category
- Widget
- Settings
- Logout

**Main Content Area:**

## Dashboard

Total Orders: 120    Revenue: \$15,000    Products: 350    Users: 500

### Manage Products

ID	Product	Price	Stock	Action
02a65e9e-4ac3-4333-a29d-438bd9f2c50c	High-Top Sneakers	65.0	80	Edit Delete
15d7fc25-93b3-45e7-b33e-18a9d0f5c6f2	Whitening Facial Bleach	20.0	200	Edit Delete
1b663720-6580-4319-b41b-89cf19503999	Silk Scarf	50.0	80	Edit Delete
1b8464bc-23a5-4933-b3b5-3c2c727ceda5	Kids Rain Boots	30.0	120	Edit Delete
1c72d153-1fe7-4a1f-b0b0-2337128f6472	Cargo Shorts	36.0	120	Edit Delete
1f27c7d9-2630-4f5f-b6e6-b2c1a2d9c3c7	Tape-In Hair Extensions	120.0	100	Edit Delete

# Product DashBoard

```
File Edit Selection View Go Run Terminal Help ← → ⌘ fash-in admin.html product.html M
src > main > resources > templates > admin > product.html > html > body > div.admin-container > main.main-content > table > tbody > tr > td.td > a.button
1  <!DOCTYPE html>
2  <html lang="en">
3  ...
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8      <link rel="stylesheet" href="/style/admin/product.css">
9      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.1/css/all.min.css">
10 </head>
11 ...
12 <body>
13     <div class="admin-container">
14         <div data-th-replace="~{'fragment/common' :: sidebar}"></div>
15 ...
16     <main class="main-content">
17         <h1>Dashboard</h1>
18         <div data-th-replace="~{'fragment/common' :: admin-cards}"></div>
19         <h2>Manage Products</h2>
20         <button id="addproductbtn">Add Product</button>
21         <table>
22             <thead>
23                 <tr>
24                     <th>Product Title</th>
25                     <th>Product Description</th>
26                     <th>Images</th>
27                     <th>Price</th>
28                     <th>Stock</th>
29                     <th>Product Status</th>
30                     <th>Action</th>
31                 </tr>
32             </thead>
33             <tbody>
34                 <tr data-th-each="product:${products}">
35                     <td data-th-text="${product.productTitle}"></td>
36                     <td data-th-text="${product.productDescription}"></td>
37                     <td></td>
38                     <td class="td" data-th-text="${product.productPrice}"></td>
39                     <td class="td" data-th-text="${product.productStockQuantity}"></td>
40                     <td data-th-text="${product.productStatus}"></td>
41                     <td class="td">
42                         <button class="button" th:attr="onclick=openModal('${product.productId}')">Edit</button>
43                         <a class="button" data-th-href="|/admin/product/remove?id=${product.productId}"> Delete</a>
44                     </td>
45                 </tr>
46             </tbody>
47         </table>
48     </main>
49     </div>
50 ...
51 </body>
52 </html>
53 ...
54 <div class="popup-overlay" id="popupOverlay">
55     <div class="popup">
56         <h2>Add Product</h2>
57         <form action="/admin/product" method="post" enctype="multipart/form-data" id="productFormModal">
58             <label for="productTitle">Product Name:</label>
59             <input type="text" placeholder="Product Name" name="productTitle" id="productTitle">
60 ...
61             <label for="productPrice">Product Price:</label>
62             <input type="text" placeholder="Product Price" name="productPrice" id="productPrice">
63 ...
64             <label for="productDescription">Product Description:</label>
65             <textarea placeholder="Product Description" name="productDescription" id="productDescription"></textarea>
66 ...
67             <label for="productImage">Product Image:</label>
68             <input type="file" name="productImageFile" accept="image/*" id="productImage">
69 ...
70             <div id="categorybutton">
71                 <select name="categoryId" id="categoryId">
72                     <option th:each="category : ${categories}" th:value="${category.categoryId}" th:text="${category.categoryTitle}"></option>
73                 </select>
74 ...
75             <button type="submit" id="submit-button">Submit</button>
76             <input type="hidden" name="productId" id="productId">
77             <input type="hidden" name="_csrf" data-th-value="${_csrf.token}">
78         </div>
79     </form>
80 ...
81     <button class="close-btn" id="closebtn">Close</button>
82 ...
83 </div>
84 ...
85 </div>
86 ...
87 <script>
```

```
File Edit Selection View Go Run Terminal Help ← → ⌘ fash-in admin.html product.html M
src > main > resources > templates > admin > product.html > html > body > div.admin-container > main.main-content > table > tbody > tr > td.td > a.button
2  <html lang="en">
3  ...
4  <body>
5      <div class="admin-container">
6          <main class="main-content">
7              <table>
8                  <tr>
9                      </tr>
10                 </tbody>
11             </table>
12         </main>
13     </div>
14 ...
15 </body>
16 </html>
17 ...
18 <div class="popup-overlay" id="popupOverlay">
19     <div class="popup">
20         <h2>Add Product</h2>
21         <form action="/admin/product" method="post" enctype="multipart/form-data" id="productFormModal">
22             <label for="productTitle">Product Name:</label>
23             <input type="text" placeholder="Product Name" name="productTitle" id="productTitle">
24 ...
25             <label for="productPrice">Product Price:</label>
26             <input type="text" placeholder="Product Price" name="productPrice" id="productPrice">
27 ...
28             <label for="productDescription">Product Description:</label>
29             <textarea placeholder="Product Description" name="productDescription" id="productDescription"></textarea>
30 ...
31             <label for="productImage">Product Image:</label>
32             <input type="file" name="productImageFile" accept="image/*" id="productImage">
33 ...
34             <div id="categorybutton">
35                 <select name="categoryId" id="categoryId">
36                     <option th:each="category : ${categories}" th:value="${category.categoryId}" th:text="${category.categoryTitle}"></option>
37                 </select>
38 ...
39             <button type="submit" id="submit-button">Submit</button>
40             <input type="hidden" name="productId" id="productId">
41             <input type="hidden" name="_csrf" data-th-value="${_csrf.token}">
42         </div>
43     </form>
44 ...
45     <button class="close-btn" id="closebtn">Close</button>
46 ...
47 </div>
48 ...
49 </div>
50 ...
51 </div>
52 ...
53 </div>
54 ...
55 </div>
56 ...
57 </div>
58 ...
59 </div>
60 ...
61 </div>
62 ...
63 </div>
64 ...
65 </div>
66 ...
67 </div>
68 ...
69 </div>
70 ...
71 </div>
72 ...
73 </div>
74 ...
75 </div>
76 ...
77 </div>
78 ...
79 </div>
80 ...
81 </div>
82 ...
83 </div>
84 ...
85 </div>
86 ...
87 </div>
88 ...
89 </div>
90 ...
91 </div>
92 ...
93 </div>
94 ...
95 </div>
96 ...
97 </div>
98 ...
99 </div>
100 ...
101 </div>
102 ...
103 </div>
104 ...
105 </div>
106 ...
107 </div>
108 ...
109 </div>
110 ...
111 </div>
112 ...
113 </div>
114 ...
115 </div>
116 ...
117 </div>
118 ...
119 </div>
120 ...
121 </div>
122 ...
123 </div>
124 ...
125 </div>
126 ...
127 </div>
128 ...
129 </div>
130 ...
131 </div>
132 ...
133 </div>
134 ...
135 </div>
136 ...
137 </div>
138 ...
139 </div>
140 ...
141 </div>
142 ...
143 </div>
144 ...
145 </div>
146 ...
147 </div>
148 ...
149 </div>
150 ...
151 </div>
152 ...
153 </div>
154 ...
155 </div>
156 ...
157 </div>
158 ...
159 </div>
160 ...
161 </div>
162 ...
163 </div>
164 ...
165 </div>
166 ...
167 </div>
168 ...
169 </div>
170 ...
171 </div>
172 ...
173 </div>
174 ...
175 </div>
176 ...
177 </div>
178 ...
179 </div>
180 ...
181 </div>
182 ...
183 </div>
184 ...
185 </div>
186 ...
187 </div>
188 ...
189 </div>
190 ...
191 </div>
192 ...
193 </div>
194 ...
195 </div>
196 ...
197 </div>
198 ...
199 </div>
200 ...
201 </div>
202 ...
203 </div>
204 ...
205 </div>
206 ...
207 </div>
208 ...
209 </div>
210 ...
211 </div>
212 ...
213 </div>
214 ...
215 </div>
216 ...
217 </div>
218 ...
219 </div>
220 ...
221 </div>
222 ...
223 </div>
224 ...
225 </div>
226 ...
227 </div>
228 ...
229 </div>
230 ...
231 </div>
232 ...
233 </div>
234 ...
235 </div>
236 ...
237 </div>
238 ...
239 </div>
240 ...
241 </div>
242 ...
243 </div>
244 ...
245 </div>
246 ...
247 </div>
248 ...
249 </div>
250 ...
251 </div>
252 ...
253 </div>
254 ...
255 </div>
256 ...
257 </div>
258 ...
259 </div>
260 ...
261 </div>
262 ...
263 </div>
264 ...
265 </div>
266 ...
267 </div>
268 ...
269 </div>
270 ...
271 </div>
272 ...
273 </div>
274 ...
275 </div>
276 ...
277 </div>
278 ...
279 </div>
280 ...
281 </div>
282 ...
283 </div>
284 ...
285 </div>
286 ...
287 </div>
288 ...
289 </div>
290 ...
291 </div>
292 ...
293 </div>
294 ...
295 </div>
296 ...
297 </div>
298 ...
299 </div>
300 ...
301 </div>
302 ...
303 </div>
304 ...
305 </div>
306 ...
307 </div>
308 ...
309 </div>
310 ...
311 </div>
312 ...
313 </div>
314 ...
315 </div>
316 ...
317 </div>
318 ...
319 </div>
320 ...
321 </div>
322 ...
323 </div>
324 ...
325 </div>
326 ...
327 </div>
328 ...
329 </div>
330 ...
331 </div>
332 ...
333 </div>
334 ...
335 </div>
336 ...
337 </div>
338 ...
339 </div>
340 ...
341 </div>
342 ...
343 </div>
344 ...
345 </div>
346 ...
347 </div>
348 ...
349 </div>
350 ...
351 </div>
352 ...
353 </div>
354 ...
355 </div>
356 ...
357 </div>
358 ...
359 </div>
360 ...
361 </div>
362 ...
363 </div>
364 ...
365 </div>
366 ...
367 </div>
368 ...
369 </div>
370 ...
371 </div>
372 ...
373 </div>
374 ...
375 </div>
376 ...
377 </div>
378 ...
379 </div>
380 ...
381 </div>
382 ...
383 </div>
384 ...
385 </div>
386 ...
387 </div>
388 ...
389 </div>
390 ...
391 </div>
392 ...
393 </div>
394 ...
395 </div>
396 ...
397 </div>
398 ...
399 </div>
400 ...
401 </div>
402 ...
403 </div>
404 ...
405 </div>
406 ...
407 </div>
408 ...
409 </div>
410 ...
411 </div>
412 ...
413 </div>
414 ...
415 </div>
416 ...
417 </div>
418 ...
419 </div>
420 ...
421 </div>
422 ...
423 </div>
424 ...
425 </div>
426 ...
427 </div>
428 ...
429 </div>
430 ...
431 </div>
432 ...
433 </div>
434 ...
435 </div>
436 ...
437 </div>
438 ...
439 </div>
440 ...
441 </div>
442 ...
443 </div>
444 ...
445 </div>
446 ...
447 </div>
448 ...
449 </div>
450 ...
451 </div>
452 ...
453 </div>
454 ...
455 </div>
456 ...
457 </div>
458 ...
459 </div>
460 ...
461 </div>
462 ...
463 </div>
464 ...
465 </div>
466 ...
467 </div>
468 ...
469 </div>
470 ...
471 </div>
472 ...
473 </div>
474 ...
475 </div>
476 ...
477 </div>
478 ...
479 </div>
480 ...
481 </div>
482 ...
483 </div>
484 ...
485 </div>
486 ...
487 </div>
488 ...
489 </div>
490 ...
491 </div>
492 ...
493 </div>
494 ...
495 </div>
496 ...
497 </div>
498 ...
499 </div>
500 ...
501 </div>
502 ...
503 </div>
504 ...
505 </div>
506 ...
507 </div>
508 ...
509 </div>
510 ...
511 </div>
512 ...
513 </div>
514 ...
515 </div>
516 ...
517 </div>
518 ...
519 </div>
520 ...
521 </div>
522 ...
523 </div>
524 ...
525 </div>
526 ...
527 </div>
528 ...
529 </div>
530 ...
531 </div>
532 ...
533 </div>
534 ...
535 </div>
536 ...
537 </div>
538 ...
539 </div>
540 ...
541 </div>
542 ...
543 </div>
544 ...
545 </div>
546 ...
547 </div>
548 ...
549 </div>
550 ...
551 </div>
552 ...
553 </div>
554 ...
555 </div>
556 ...
557 </div>
558 ...
559 </div>
560 ...
561 </div>
```

A screenshot of a code editor interface. The main area displays the file `product.html`. The code is a combination of HTML and JavaScript. It includes a modal overlay for adding products, event listeners for buttons, and an asynchronous function to fetch product data from a server. The status bar at the bottom shows indexing completed and Java Ready.

```
src > main > resources > templates > admin > product.html > html > body > div.admin-container > main.main-content > table > tbody > tr > td.td > a.button
2  <html lang="en">
3  <body>
4      <div class="popup-overlay" id="popupOverlay">
5          <div class="popup">
6              <form action="/admin/product" method="post" enctype="multipart/form-data" id="productFormModal">
7                  <input type="hidden" name="_csrf" data-th-value="${_csrf.token}">
8              </div>
9          </form>
10         <button class="close-btn" id="closebtn">Close</button>
11     </div>
12   </div>
13
14 <script>
15     let displayDiv = document.getElementById("popupOverlay");
16     let closeButton = document.getElementById("closebtn");
17     let submitButton = document.getElementById("submit-button");
18     let button = document.getElementById("addproductbtn");
19     button.addEventListener('click', () => {
20         displayDiv.style.display = "block";
21     });
22
23     closeButton.addEventListener('click', () => {
24         displayDiv.style.display = "none";
25     });
26
27     submitButton.addEventListener('click', () => {
28         displayDiv.style.display = "none";
29     });
30
31     async function openModal(productId = null) {
32         displayDiv.style.display = "block";
33         if (productId) {
34             const response = await fetch('/admin/products/${productId}').catch((err) =>
35                 console.error(err)
36             );
37
38             const product = await response.json();
39             updateFields(product);
40         } else {
41             document.getElementById('productFormModal').reset();
42         }
43     }
44
45     function deleteProduct (element){
46         element.parentElement.parentElement.remove();
47     }
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
```

A screenshot of a code editor interface. The main area displays the file `product.html`. The code is identical to the one in the previous screenshot, showing the modal overlay, event listeners, and asynchronous fetch function. The status bar at the bottom shows indexing completed and Java Ready.

```
src > main > resources > templates > admin > product.html > html > body > script
2  <html lang="en">
3  <body>
4      <script>
5          button.addEventListener('click', () => {
6              displayDiv.style.display = "block";
7          });
8
9          closeButton.addEventListener('click', () => {
10             displayDiv.style.display = "none";
11         });
12
13         submitButton.addEventListener('click', () => {
14             displayDiv.style.display = "none";
15         });
16
17
18         async function openModal(productId = null) {
19             displayDiv.style.display = "block";
20             if (productId) {
21                 const response = await fetch('/admin/products/${productId}').catch((err) =>
22                     console.error(err)
23                 );
24
25                 const product = await response.json();
26                 updateFields(product);
27             } else {
28                 document.getElementById('productFormModal').reset();
29             }
30         }
31
32         function deleteProduct (element){
33             element.parentElement.parentElement.remove();
34         }
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
109
110
111
112
113
114
115
116
117
118
119
119
120
121
122
123
124
125
126
127
128
129
129
130
131
132
133
134
135
```

## Fashin Admin

Dashboard

Orders

Products

Category

Widget

Settings

Logout

## Dashboard

Total Orders: 120

Revenue: \$15,000

Products: 350

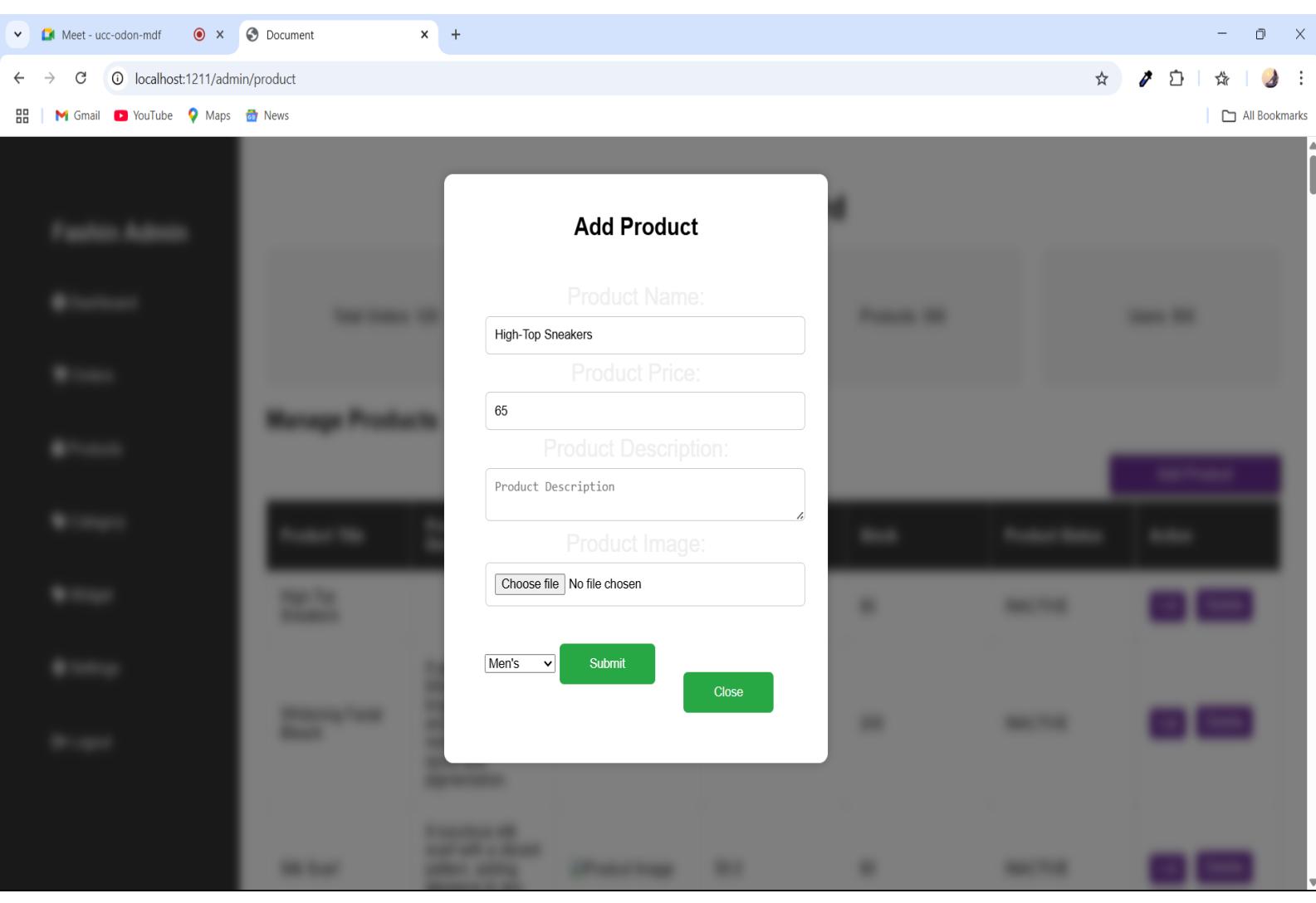
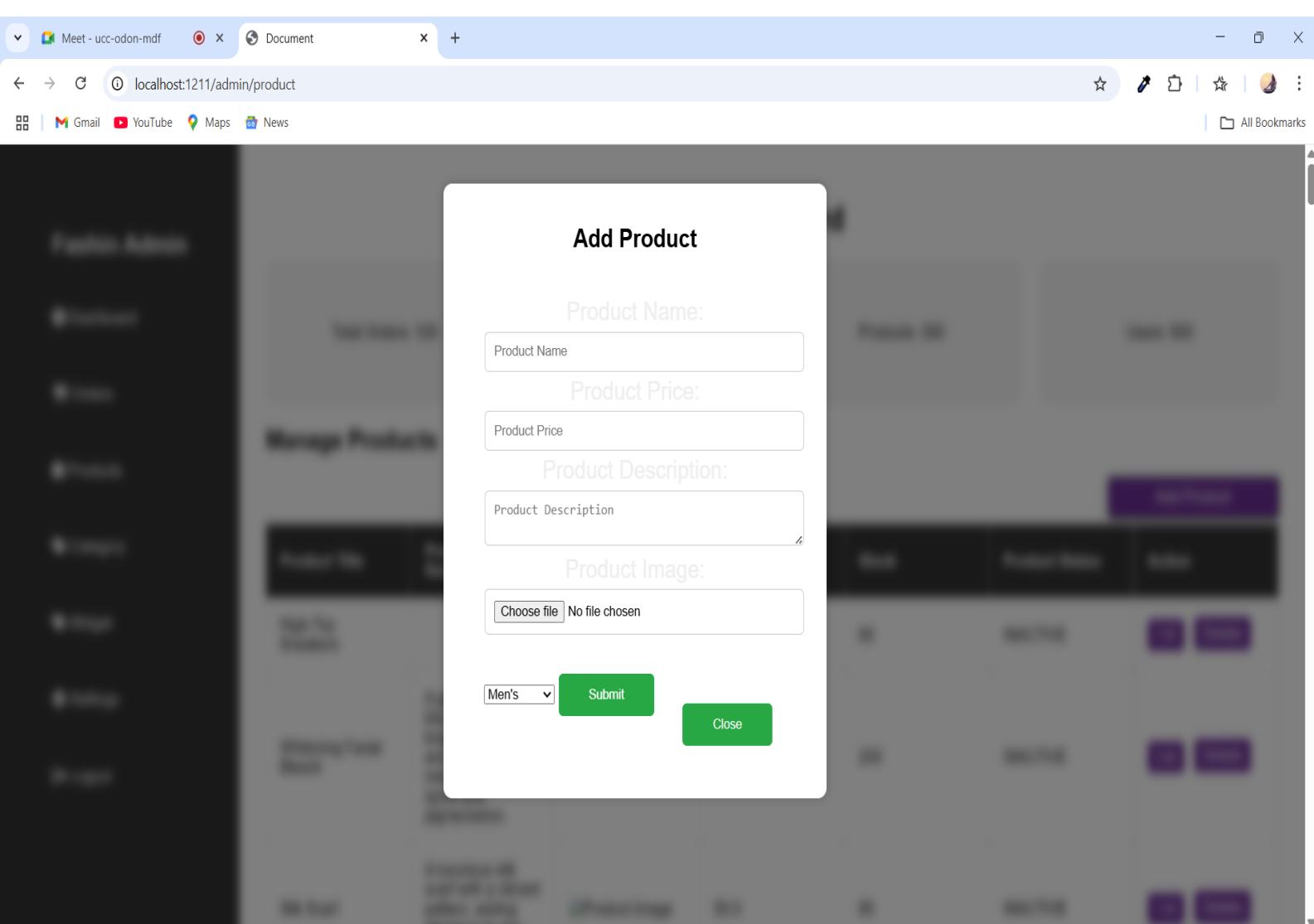
Users: 500

## Manage Products

Add Product

Product Title	Product Description	Images	Price	Stock	Product Status	Action
High-Top Sneakers			65.0	80	INACTIVE	<button>Edit</button> <button>Delete</button>
Whitening Facial Bleach	A gentle facial bleach that brightens the skin and helps in reducing dark spots and pigmentation.		20.0	200	INACTIVE	<button>Edit</button> <button>Delete</button>
Silk Scarf	A luxurious silk scarf with a vibrant pattern, adding elegance to any		50.0	80	INACTIVE	<button>Edit</button> <button>Delete</button>

Cargo Shorts	Stylish cargo shorts with multiple pockets, perfect for outdoor activities.		36.0	120	INACTIVE	<button>Edit</button> <button>Delete</button>
Tape-In Hair Extensions	Premium tape-in hair extensions that seamlessly blend with your natural hair for a fuller look.		120.0	100	ACTIVE	<button>Edit</button> <button>Delete</button>
Whitening Facial Bleach	A gentle facial bleach that brightens the skin and helps in reducing dark spots and pigmentation. Yes		20.0	0	ACTIVE	<button>Edit</button> <button>Delete</button>
Facial Roller	A facial roller designed to stimulate blood circulation, reduce puffiness, and promote relaxation.		20.0	150	ACTIVE	<button>Edit</button> <button>Delete</button>
soumendra	student of jt		4568.0	0	ACTIVE	<button>Edit</button> <button>Delete</button>
Oversized Sunglasses	Trendy oversized sunglasses, a must-have accessory to complete any		40.0	200	ACTIVE	<button>Edit</button> <button>Delete</button>



# Category on admin DashBoard

```
File Edit Selection View Go Run Terminal Help ← → ⌘ fash-in
admin.html category.html
src > main > resources > templates > admin > category.html > html > body > div.admin-container > main.main-content > table
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8      <link rel="stylesheet" href="/style/category.css">
9      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.1/css/all.min.css">
10 </head>
11
12 <body>
13     <div class="admin-container">
14         <div data-th-replace="{'fragment/common' :: sidebar}"></div>
15
16         <main class="main-content">
17             <h1>Dashboard</h1>
18             <div data-th-replace="{'fragment/common' :: admin-cards}"></div>
19             <h2>Manage Category</h2>
20
21             <table>
22                 <thead>
23                     <tr>
24                         <th>Category ID</th>
25                         <th>Category</th>
26                         <th>Stock</th>
27                     </tr>
28                 </thead>
29                 <tbody>
30                     <tr data-th-each="category:${categories}">
31                         <td data-th-text="${category.categoryId}"></td>
32                         <td data-th-text="${category.categoryTitle}"></td>
33                         <!-- <td></td> -->
34                         <td class="td" data-th-text="${category.categoryStatus}">65</td>
35                     </tr>
36                 </tbody>
37             </table>
38         </main>
39     </div>
40
41     <form class="popup-overlay" id="popupOverlay">
42         <div class="popup">
43             <h2>Add Category</h2>
44             <label for="categoryName">Category Name:</label>
45             <input type="text" placeholder="Category Name">
46
47             <label for="categoryId">Category ID:</label>
48             <input type="text" placeholder="Category ID">
49
50             <label for="categoryPrice">Category Price:</label>
51             <input type="text" placeholder="Category Price">
52
53             <label for="categoryDescription">Category Description:</label>
54             <textarea placeholder="Category Description"></textarea>
55
56             <label for="categoryImage">Category Image:</label>
57             <input type="file" accept="image/*">
58
59             <button id="submit-button">Submit</button>
60             <button class="close-btn" id="closebtn">Close</button>
61         </div>
62     </form>
63
64
65 </html>
```

Indexing completed. Java Ready

```
File Edit Selection View Go Run Terminal Help ← → ⌘ fash-in
admin.html category.html
src > main > resources > templates > admin > category.html > html > body > div.admin-container > main.main-content > table
2  <html lang="en">
3
4  <div class="admin-container">
5      <main class="main-content">
6          <table>
7              <thead>
8                  <tr>
9                      <td data-th-text="${category.categoryId}"></td>
10                     <td data-th-text="${category.categoryTitle}"></td>
11                     <!-- <td></td> -->
12                     <td class="td" data-th-text="${category.categoryStatus}">65</td>
13                 </tr>
14             </thead>
15             <tbody>
16                 <tr>
17                     <td data-th-text="${category.categoryId}"></td>
18                     <td data-th-text="${category.categoryTitle}"></td>
19                     <td></td>
20                     <td class="td" data-th-text="${category.categoryStatus}">65</td>
21                 </tr>
22             </tbody>
23         </table>
24     </main>
25
26     <form class="popup-overlay" id="popupOverlay">
27         <div class="popup">
28             <h2>Add Category</h2>
29             <label for="categoryName">Category Name:</label>
30             <input type="text" placeholder="Category Name">
31
32             <label for="categoryId">Category ID:</label>
33             <input type="text" placeholder="Category ID">
34
35             <label for="categoryPrice">Category Price:</label>
36             <input type="text" placeholder="Category Price">
37
38             <label for="categoryDescription">Category Description:</label>
39             <textarea placeholder="Category Description"></textarea>
40
41             <label for="categoryImage">Category Image:</label>
42             <input type="file" accept="image/*">
43
44             <button id="submit-button">Submit</button>
45             <button class="close-btn" id="closebtn">Close</button>
46         </div>
47     </form>
48
49
50 </html>
```

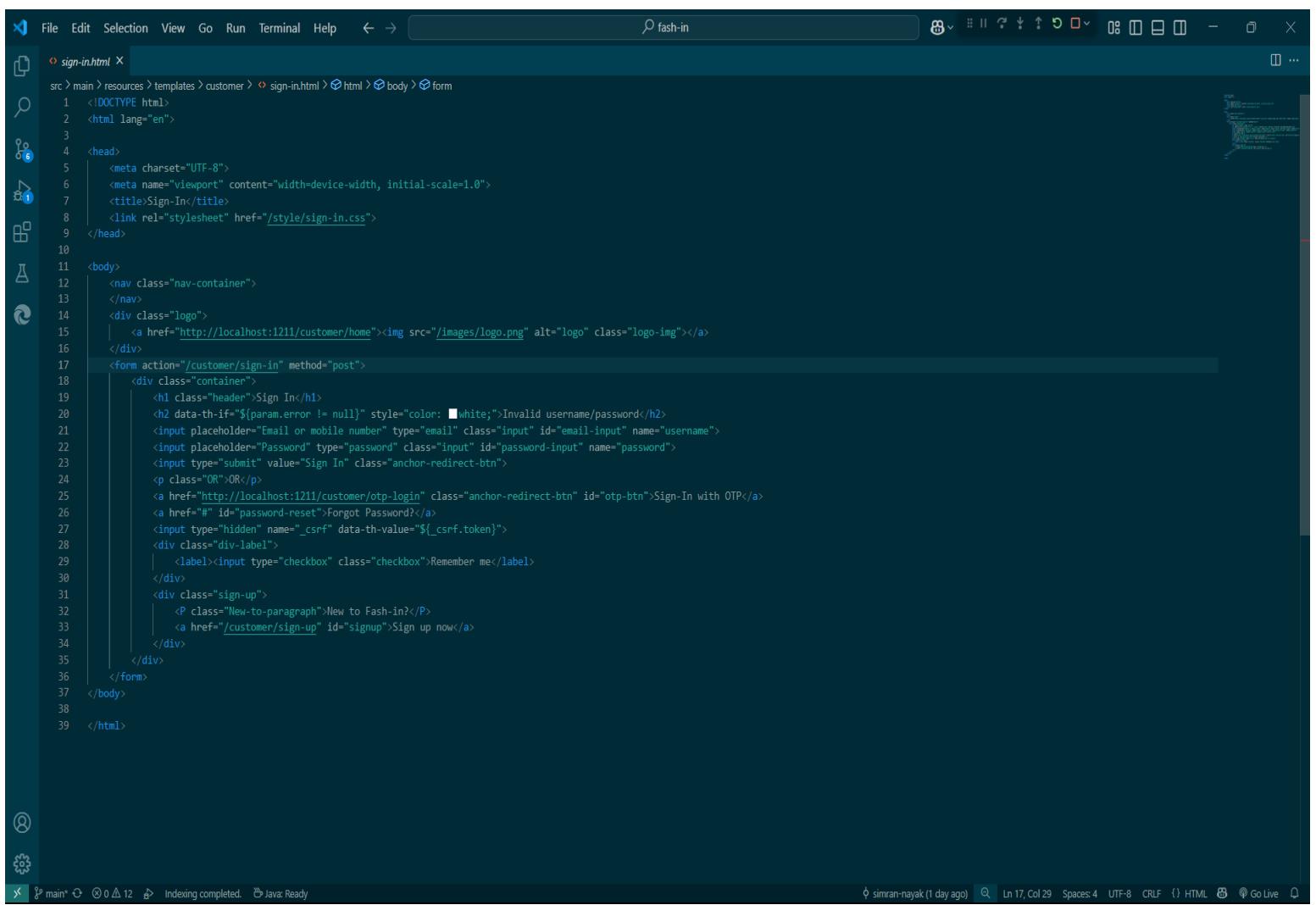
Indexing completed. Java Ready

# SnapShot of Category on Admin Dashboard

The screenshot shows the Fashin Admin dashboard. On the left is a dark sidebar with navigation links: Dashboard, Orders, Products, Category, Widget, Settings, and Logout. The main area has a title "Dashboard" and four summary cards: Total Orders: 120, Revenue: \$15,000, Products: 350, and Users: 500. Below this is a section titled "Manage Category" containing a table with eight rows of category data.

Category ID	Category	Stock
a1234567-89ab-4cde-1234-56789abcdef0	Men's	ACTIVE
b1234567-89ab-4cde-1234-56789abcdef1	Women's	ACTIVE
c1234567-89ab-4cde-1234-56789abcdef2	Kid's	ACTIVE
d1234567-89ab-4cde-1234-56789abcdef3	Fragrance	ACTIVE
e1234567-89ab-4cde-1234-56789abcdef4	Skincare	ACTIVE
f1234567-89ab-4cde-1234-56789abcdef5	Haircare	ACTIVE
g1234567-89ab-4cde-1234-56789abcdef6	Beauty	ACTIVE

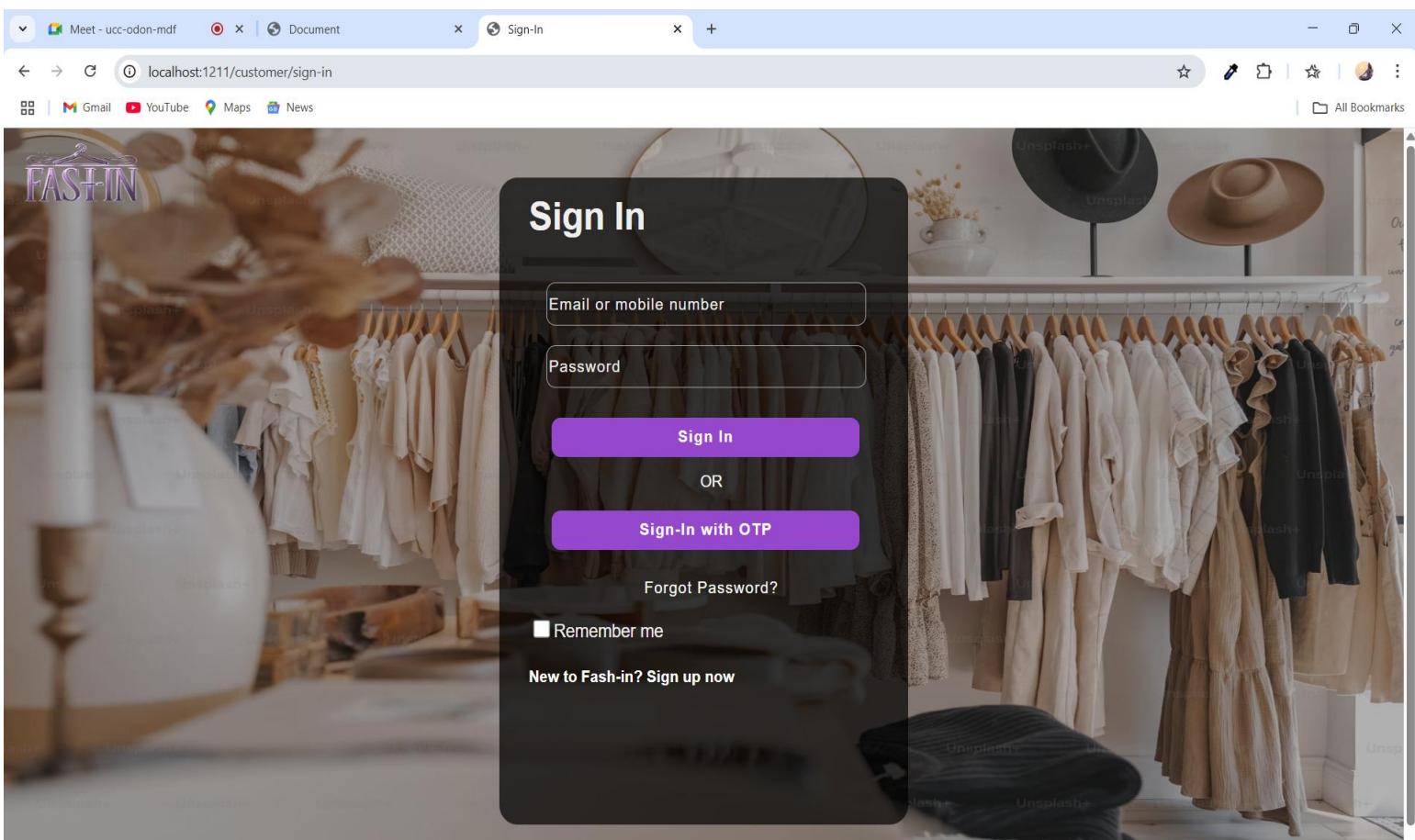
# Sign In



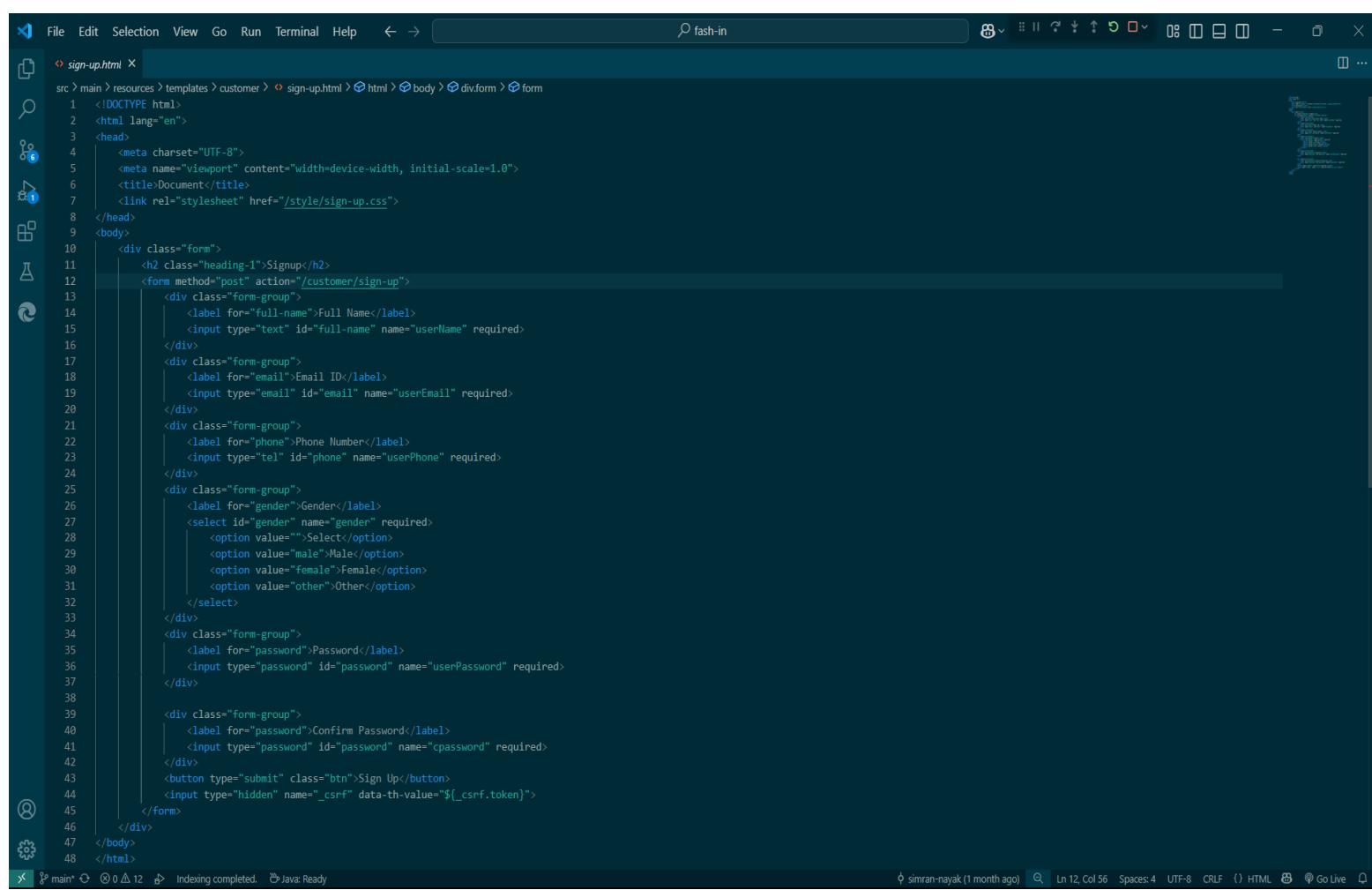
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Sign-In</title>
        <link rel="stylesheet" href="/style/sign-in.css">
    </head>
    <body>
        <nav class="nav-container">
            </nav>
        <div class="logo">
            <a href="http://localhost:1211/customer/home"></a>
        </div>
        <form action="/customer/sign-in" method="post">
            <div class="container">
                <h1 class="header">Sign In</h1>
                <h2 data-th-if="${param.error != null}" style="color: #white;">Invalid username/password</h2>
                <input placeholder="Email or mobile number" type="email" class="input" id="email-input" name="username">
                <input placeholder="Password" type="password" class="input" id="password-input" name="password">
                <input type="submit" value="Sign In" class="anchor-redirect-btn">
                <p class="OR">OR</p>
                <a href="http://localhost:1211/customer/otp-login" class="anchor-redirect-btn" id="otp-btn">Sign-In with OTP</a>
                <a href="#" id="password-reset">Forgot Password?</a>
                <input type="hidden" name="_csrf" data-th-value="${_csrf.token}">
                <div class="div-label">
                    <label><input type="checkbox" class="checkbox">Remember me</label>
                </div>
                <div class="sign-up">
                    <p class="New-to-paragraph">New to Fash-in?</p>
                    <a href="/customer/sign-up" id="signup">Sign up now</a>
                </div>
            </div>
        </form>
    </body>
</html>
```

⌚ simran-nayak (1 day ago) 🔍 Ln 17, Col 29 Spaces: 4 UTF-8 CRLF { HTML 🌐 Go Live 🚀

## SnapShot Of SignIn



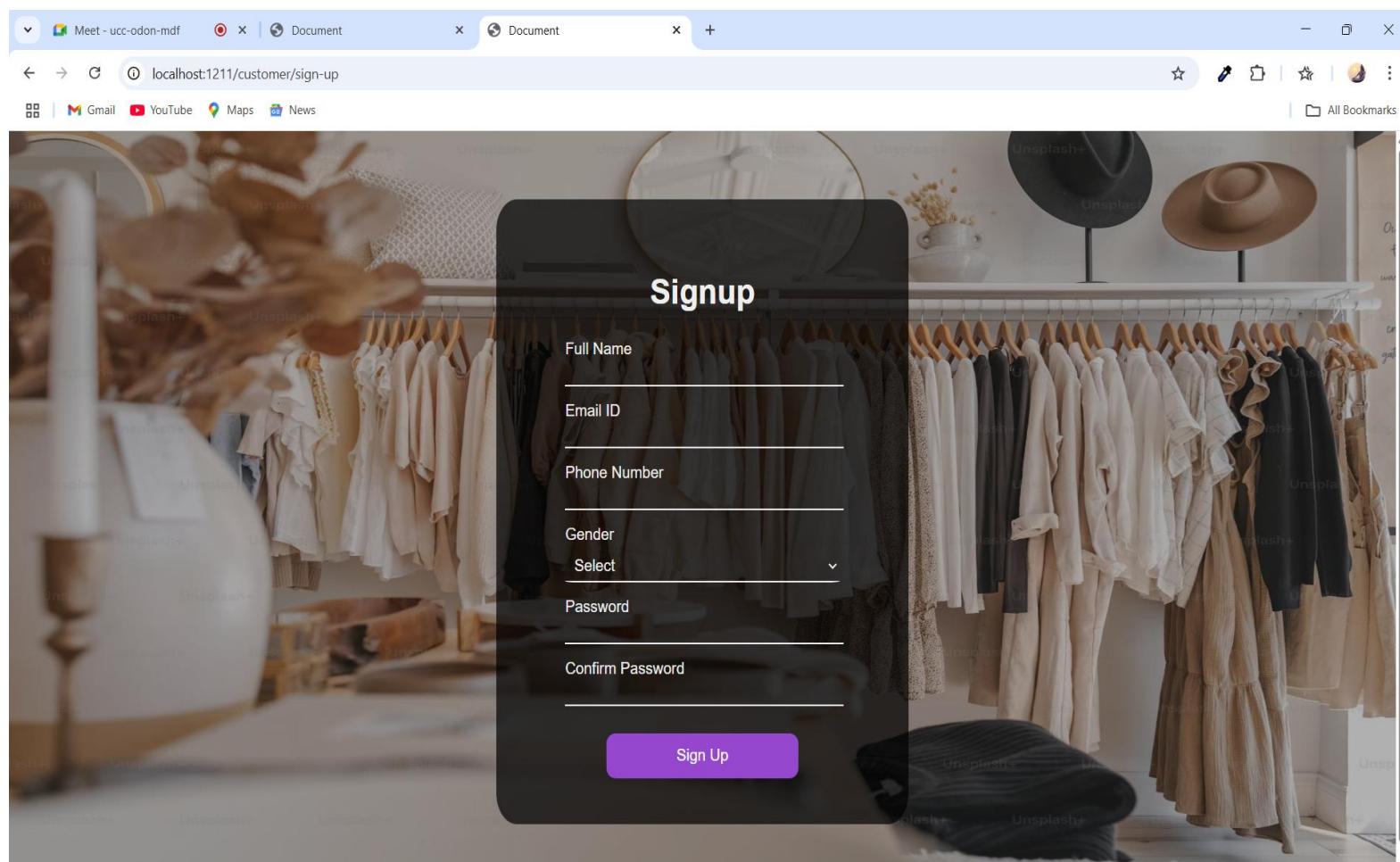
# Sign Up



A screenshot of a code editor showing the content of a file named "sign-up.html". The code is an HTML form for user registration, including fields for full name, email, phone number, gender, password, and confirmation. It uses CSS classes like "form-group" and "heading-1". A CSRF token input field is also present.

```
src > main > resources > templates > customer > sign-up.html > HTML > body > div.form > form
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7      <link rel="stylesheet" href="/style/sign-up.css">
8  </head>
9  <body>
10     <div class="form">
11         <h2 class="heading-1">Signup</h2>
12         <form method="post" action="/customer/sign-up">
13             <div class="form-group">
14                 <label for="full-name">Full Name</label>
15                 <input type="text" id="full-name" name="userName" required>
16             </div>
17             <div class="form-group">
18                 <label for="email">Email ID</label>
19                 <input type="email" id="email" name="userEmail" required>
20             </div>
21             <div class="form-group">
22                 <label for="phone">Phone Number</label>
23                 <input type="tel" id="phone" name="userPhone" required>
24             </div>
25             <div class="form-group">
26                 <label for="gender">Gender</label>
27                 <select id="gender" name="gender" required>
28                     <option value="">Select</option>
29                     <option value="male">Male</option>
30                     <option value="female">Female</option>
31                     <option value="other">Other</option>
32                 </select>
33             </div>
34             <div class="form-group">
35                 <label for="password">Password</label>
36                 <input type="password" id="password" name="userPassword" required>
37             </div>
38
39             <div class="form-group">
40                 <label for="password">Confirm Password</label>
41                 <input type="password" id="cpassword" name="cpassword" required>
42             </div>
43             <button type="submit" class="btn">Sign Up</button>
44             <input type="hidden" name="_csrf" data-th-value="${_csrf.token}">
45         </form>
46     </div>
47 </body>
48 </html>
```

## SnapShot of SignUp



# Cart Page

File Edit Selection View Go Run Terminal Help ⏪ ⏩ fash-in

Cart.html

```
src > main > resources > templates > customer > Cart.html > html > body > div#addressModal.modal > div.modal-content > h2
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Cart</title>
8      <link rel="stylesheet" href="/style/cart.css">
9      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">
10 </head>
11
12 <body>
13     <div class="navbar">
14         <div class="logo">
15             |     <a href="#"></a>
16         </div>
17         <div class="search-container">
18             |     <input type="text" class="search-input" placeholder="Search your Items .....">
19             |     <i class="search-icon fas fa-search"></i>
20         </div>
21
22         <div class="dropdown">
23             <button class="dropbtn">Profile #9660;</button>
24             <div class="dropdown-content">
25                 |     <a href="#">My Profile</a>
26                 |     <a href="#">Order</a>
27                 |     <a href="#">Wishlist</a>
28                 |     <a href="#">Coupons</a>
29                 |     <a href="#">Gift Cards</a>
30                 |     <a href="#">Notifications</a>
31                 |     <a href="#" id="Logout">Logout</a>
32             </div>
33         </div>
34     </div>
35
36     <div class="main">
37         <div class="wrap">
38             |     <span class="address-reference">To your Current Saved address </span>
39             |     <button class="address-update" onclick="openModal()">Change address</button>
40         </div>
41
42         <div class="dropdown-container1">
43             <label for="address-dropdown">Select Delivery Address:</label>
44             <select id="address-dropdown" class="address-dropdown">
45                 |     <option value="home">Home - 123 Main Street</option>
46                 |     <option value="office">Office - 456 Business Ave</option>
47                 |     <option value="other">Other</option>
48             </select>
49
50
51     </div>
52
53     <div class="container">
54         <div class="wrapper">
55             |     
56             |     <div class="info">
57                 |         <h3>Men Regular Fit Checked Spread Collar Casual Shirt</h3>
58                 |         <p class="cart-specs">Fits: Regular | Size: XL</p>
59                 |         <span class="deliver-date">Delivery by Sun Feb 23 | <span class="charges-span">$40</span><span class="free-span">Free</span></span>
60                 |         <span class="actual-price">$176</span>
61                 |         <span class="amount" id="amount">$769</span>
62                 |         <span class="offer">63% off 2 Offers available !!</span>
63             </div>
64
65         </div>
66
67         <div class="counter-box">
68             <button onclick="decrease()">-</button>
69             <div class="count" id="count">0</div>
70             <button onclick="increase()">+</button>
71         </div>
72
73         <div class="place-order">
74             |     <button class="order-btn">Place Order</button>
75         </div>
76     </div>
77
78     <div class="Calc">
79         <h3>Price Details</h3>
80         <hr>
81         <div class="amount" id="displayprice">Price:</div>
82         <div class="discount">Discount:</div>
83         <div class="delivery-charges">Delivery Charges:</div>
84         <hr>
85         <div class="total">Total Amount: <span id="total-amount">$0</span></div>
86         <hr>
87     </div>
88
89     <div class="dropdown-payment">
90         <label for="payment-dropdown">Select Payment Method:</label>
91         <select id="payment-dropdown" class="payment-dropdown">
92             |     <option value="card">Card</option>
93             |     <option value="cash">Cash</option>
94             |     <option value="bank">Bank Transfer</option>
95         </select>
96     </div>
97
98
99
100
```

File Edit Selection View Go Run Terminal Help ⏪ ⏩ fash-in

Cart.html

```
src > main > resources > templates > customer > Cart.html > html > body > div#addressModal.modal > div.modal-content > h2
2  <html lang="en">
3
4  <body>
5
6      <div class="main">
7          <div class="dropdown-container1">
8              </div>
9
10     <div class="container">
11         <div class="wrapper">
12             |     
13             |     <div class="info">
14                 |         <h3>Men Regular Fit Checked Spread Collar Casual Shirt</h3>
15                 |         <p class="cart-specs">Fits: Regular | Size: XL</p>
16                 |         <span class="deliver-date">Delivery by Sun Feb 23 | <span class="charges-span">$40</span><span class="free-span">Free</span></span>
17                 |         <span class="actual-price">$176</span>
18                 |         <span class="amount" id="amount">$769</span>
19                 |         <span class="offer">63% off 2 Offers available !!</span>
20             </div>
21
22         </div>
23
24         <div class="counter-box">
25             <button onclick="decrease()">-</button>
26             <div class="count" id="count">0</div>
27             <button onclick="increase()">+</button>
28         </div>
29
30         <div class="place-order">
31             |     <button class="order-btn">Place Order</button>
32         </div>
33     </div>
34
35     <div class="Calc">
36         <h3>Price Details</h3>
37         <hr>
38         <div class="amount" id="displayprice">Price:</div>
39         <div class="discount">Discount:</div>
40         <div class="delivery-charges">Delivery Charges:</div>
41         <hr>
42         <div class="total">Total Amount: <span id="total-amount">$0</span></div>
43         <hr>
44     </div>
45
46     <div class="dropdown-payment">
47         <label for="payment-dropdown">Select Payment Method:</label>
48         <select id="payment-dropdown" class="payment-dropdown">
49             |     <option value="card">Card</option>
50             |     <option value="cash">Cash</option>
51             |     <option value="bank">Bank Transfer</option>
52         </select>
53     </div>
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

File Edit Selection View Go Run Terminal Help ← → ⌘ fash-in

Cart.html x

```
src > main > resources > templates > customer > Cart.html > html > body > div#addressModal.modal > div.modal-content > h2
2   <html lang="en">
12  <body>
79   <div class="Calc">
83     <div class="discount">Discount:</div>
84     <div class="delivery-charges">Delivery Charges:</div>
85     <hr>
86     <div class="total">Total Amount: <span id="total-amount">$0</span></div>
87     <hr>
88   </div>
89
90   <div class="dropdown-payment">
91     <label for="payment-dropdown">Select Payment Method:</label>
92     <select id="payment-dropdown" class="payment-dropdown">
93       <option value="cod">Cash on Delivery</option>
94       <option value="credit">Credit Card</option>
95       <option value="debit">Debit Card</option>
96       <option value="upi">UPI</option>
97       <option value="paypal">PayPal</option>
98     </select>
99   </div>
100
101 <div id="addressModal" class="modal">
102   <div class="modal-content">
103     <span class="close" onclick="closeModal()">✖</span>
104     <h2>Enter New Address</h2>
105     <form id="addressForm">
106       <label for="fullName">Full Name:</label>
107       <input type="text" id="fullName" required>
108
109       <label for="phone">Phone Number:</label>
110       <input type="text" id="phone" required>
111
112       <label for="street">Street Address:</label>
113       <input type="text" id="street" required>
114
115       <label for="city">City:</label>
116       <input type="text" id="city" required>
117
118       <label for="state">State:</label>
119       <input type="text" id="state" required>
120
121       <label for="zip">Zip Code:</label>
122       <input type="text" id="zip" required>
123
124       <button type="submit">Save Address</button>
125     </form>
126   </div>
127 </div>
```

Indexing completed. Java Ready

AdityaKhandwal (4 days ago) Q Ln 104 Col 39 Spaces: 4 UTF-8 CRLF {} HTML ⚡ Port: 5500

File Edit Selection View Go Run Terminal Help ← → ⌘ fash-in

Cart.html x

```
src > main > resources > templates > customer > Cart.html > html > body > div#addressModal.modal > div.modal-content > h2
2   <html lang="en">
12  <body>
101   <div id="addressModal" class="modal">
102     <div class="modal-content">
125       </form>
126     </div>
127   </div>
128
129 <script>
130   let count = 0;
131   let itemPrice = parseFloat(document.getElementById("amount").textContent.replace("$", ""));
132   let totalAmount = 0;
133   let displayProductPrice = document.getElementById("displayprice");
134   displayProductPrice.textContent = "Price: $" + itemPrice;
135
136   function updateTotal() {
137     totalAmount = count * itemPrice;
138     document.getElementById("total-amount").textContent = "$" + totalAmount;
139   }
140
141   function increase() {
142     count++;
143     document.getElementById("count").textContent = count;
144     updateTotal();
145   }
146
147   function decrease() {
148     if (count > 0) {
149       count--;
150       document.getElementById("count").textContent = count;
151       updateTotal();
152     }
153   }
154
155   function openModal() {
156     document.getElementById("addressModal").style.display = "block";
157   }
158
159   function closeModal() {
160     document.getElementById("addressModal").style.display = "none";
161   }
162
163   window.onclick = function (event) {
164     if (event.target == document.getElementById("addressModal")) {
165       closeModal();
166     }
167   }
168 </script>
```

Indexing completed. Java Ready

AdityaKhandwal (4 days ago) Q Ln 104 Col 39 Spaces: 4 UTF-8 CRLF {} HTML ⚡ Port: 5500

# SnapShot of Cart Page

Meet - ucc-odon-mdf Document Document Cart

localhost:1211/customer/cart

Gmail YouTube Maps News All Bookmarks

FASHIN

Search your Items .....

Profile ▾

To your Current Saved address > Change address

Select Delivery Address: Home - 123 Main Street

**Men Regular Fit Checkered Spread Collar Casual Shirt** Delivery by Sun Feb 23 | \$40 Free

Fits: Regular | Size: XL

\$1766 **\$769** 63% off 2 Offers available !!

- 0 +

Place Order

Price Details

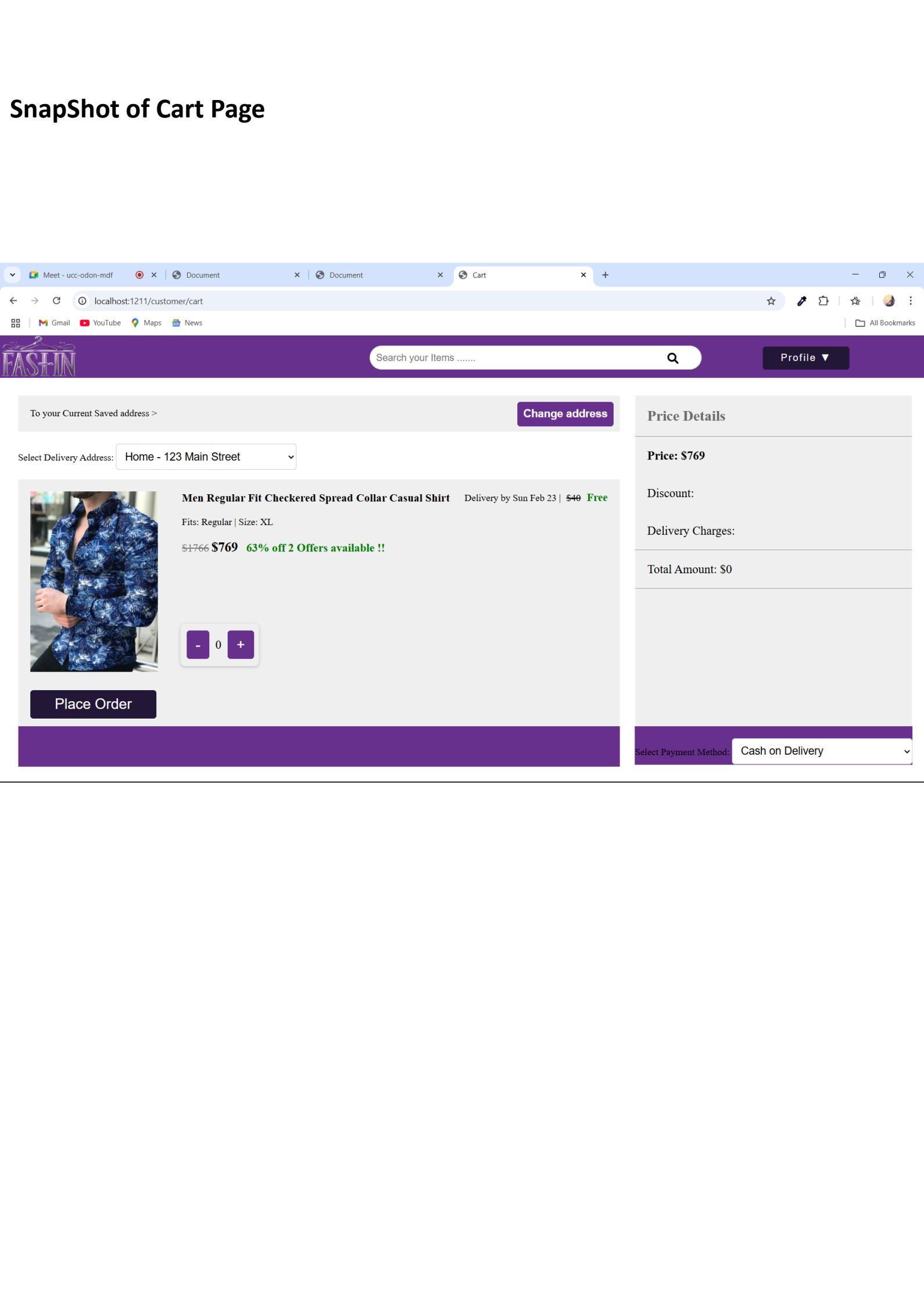
**Price: \$769**

Discount:

Delivery Charges:

Total Amount: \$0

Select Payment Method: Cash on Delivery

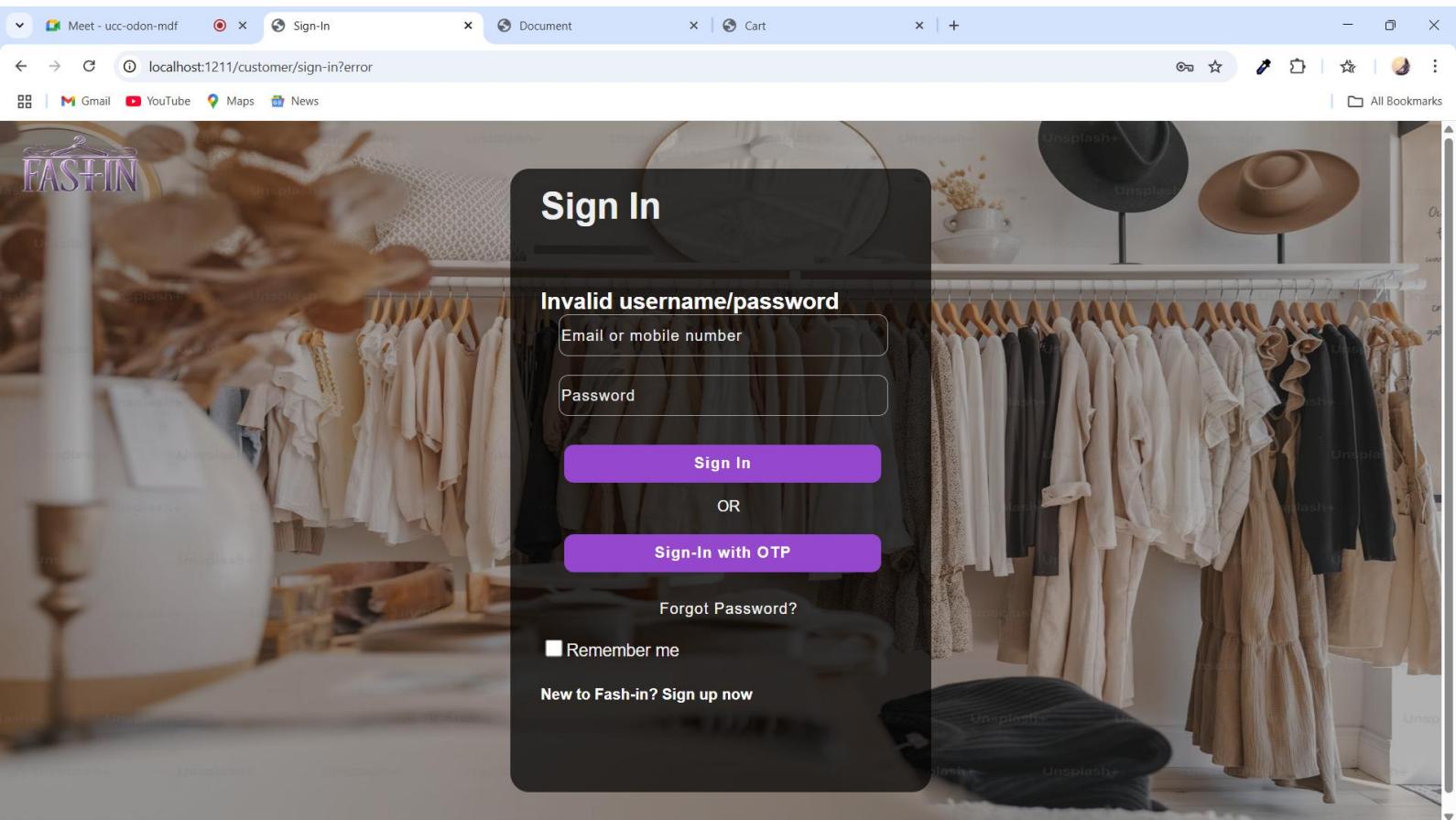


# Spring Security

```
src > main > java > edu > rims > fash_in > config > J SecurityConfig.java
1 package edu.rims.fash_in.config;
2
3 import lombok.RequiredArgsConstructor;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.context.annotation.Bean;
7 import org.springframework.context.annotation.Configuration;
8 import org.springframework.security.config.Customizer;
9 import org.springframework.security.config.annotation.web.builders.HttpSecurity;
10 import org.springframework.security.core.userdetails.UserDetailsService;
11 import org.springframework.security.core.userdetails.UsernameNotFoundException;
12 import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
13 import org.springframework.security.crypto.password.PasswordEncoder;
14 import org.springframework.security.web.SecurityFilterChain;
15
16 import edu.rims.fash_in.repository.UserRepository;
17
18
19 @Configuration
20 public class SecurityConfig {
21     @Autowired
22     private UserRepository userRepository;
23
24     @Bean
25     SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
26         http.authorizeHttpRequests(
27             request -> request
28                 .requestMatchers("/style/**", "/JavaScript/**", "/images/**", "/file/**").permitAll()
29                 .requestMatchers("/customer/home", "/customer/plp", "/customer/pdp").permitAll()
30                 .requestMatchers("/customer/sign-in", "/customer/sign-up").permitAll()
31                 .requestMatchers("/file/**", "/user/**", "/").permitAll()
32                 .requestMatchers("/admin/**").hasRole("ADMIN")
33                 .anyRequest().authenticated());
34         http.formLogin(form -> form.loginPage("/customer/sign-in")
35                         .defaultSuccessUrl("/customer/home"));
36         http.logout(Customizer.withDefaults());
37         return http.build();
38     }
39
40     @Bean
41     PasswordEncoder encoder() {
42         return new BCryptPasswordEncoder(12);
43     }
44
45     @Bean
46     UserDetailsService detailsService() {
47         return username -> {
48             var user = userRepository.findByUserEmail(username)
49                     .orElseThrow(() -> new UsernameNotFoundException("User Not Found"));
50             return org.springframework.security.core.userdetails.User.builder()
51                 .username(username)
52                 .password(user.getUserPassword())
53                 .roles(user.getUserRole().toString())
54                 .build();
55         };
56     }
57
58 }
```

```
src > main > java > edu > rims > fash_in > config > J SecurityConfig.java
1 import edu.rims.fash_in.repository.UserRepository;
2
3
4 @Configuration
5 public class SecurityConfig {
6     @Autowired
7     private UserRepository userRepository;
8
9     @Bean
10    SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
11        http.authorizeHttpRequests(
12            request -> request
13                .requestMatchers("/style/**", "/JavaScript/**", "/images/**", "/file/**").permitAll()
14                .requestMatchers("/customer/home", "/customer/plp", "/customer/pdp").permitAll()
15                .requestMatchers("/customer/sign-in", "/customer/sign-up").permitAll()
16                .requestMatchers("/file/**", "/user/**", "/").permitAll()
17                .requestMatchers("/admin/**").hasRole("ADMIN")
18                .anyRequest().authenticated());
19        http.formLogin(form -> form.loginPage("/customer/sign-in")
20                        .defaultSuccessUrl("/customer/home"));
21        http.logout(Customizer.withDefaults());
22        return http.build();
23    }
24
25    @Bean
26    PasswordEncoder encoder() {
27        return new BCryptPasswordEncoder(12);
28    }
29
30    @Bean
31    UserDetailsService detailsService() {
32        return username -> {
33            var user = userRepository.findByUserEmail(username)
34                    .orElseThrow(() -> new UsernameNotFoundException("User Not Found"));
35            return org.springframework.security.core.userdetails.User.builder()
36                .username(username)
37                .password(user.getUserPassword())
38                .roles(user.getUserRole().toString())
39                .build();
40        };
41    }
42
43 }
```

## When User give wrong id or Password



# **CHAPTER-10**

---

## **Testing**

---

### **Introduction**

The goal of software testing is to evaluate software program capabilities or attributes to ensure that it meets the application standards. Testing does not guarantee quality, and its objective is not to detect bugs. Testing might consist of verification and validation as well as dependability estimation. The fundamental goal of testing is to find errors in application. The most crucial function of testing is to provide information. Check ensure that the program is functioning properly while creating, updating, and deleting product entries.

### **Unit Testing**

Individual software components are tested in this manner. It is usually done by the programmer rather than the testers. This demands detailed information and expertise of the internal program design and code. We performed numerous testing tasks during unit testing, such as the reflection of the unit data on the database and its interface. Several sorts of bugs linked with the component were discovered and resolved. To test our software, we use a variety of functional keys. Our software unit testing is concerned with stock units, opening stock units, and product unit validation.

**Future Enhancements** To improve and expand the FashIn Web Application, the following future enhancements can be considered:

1. Advanced Search & Filtering – Implement enhanced search options with filters based on color, price range, occasion, and material to improve user experience.

2. Wishlist & Favorites – Allow users to save their favorite products for future reference, enhancing engagement and return visits.

3. Personalized Recommendations – Introduce a recommendation system based on user browsing and purchase history to improve product discovery.

4. Social Media Integration – Enable users to share their favorite fashion picks on social media platforms and log in using social accounts for a smoother experience.

5. User Reviews & Ratings – Add a review and rating system to help customers make informed purchasing decisions.

6. Loyalty & Reward Programs – Introduce a points-based system where users earn rewards for purchases, referrals, or engagement.

7. Live Chat Support – Implement real-time customer support to address user queries and provide a better shopping experience.

8. Mobile App Version – Develop a mobile application for iOS and Android to reach a broader audience and improve accessibility.

9. Augmented Reality (AR) Try-On – Integrate AR-based try-on features to help users visualize outfits and accessories before purchasing.

10. Multi-Vendor Marketplace – Expand the platform to allow multiple sellers to list their products, transforming it into a full-fledged marketplace. By implementing these enhancements, the platform can stay competitive, improve user engagement, and drive more traffic, ensuring long-term growth and success.

---

## References

---

- Google
- ChatGPT
- Java Technocrat, BBSR (<https://www.javatechnocrat.com/>)
- W3school (<https://www.w3schools.com/MySQL/default.asp>)
- GeeksforGeeks (<https://www.geeksforgeeks.org/spring/>)

---

## Conclusion and Future Scope

---

### Conclusion

The \*FashIn\* Web Application\* successfully delivers a seamless and user-friendly platform for fashion enthusiasts. By offering an intuitive interface, well-organized product categories, and a visually appealing design, the application enhances the online shopping and lifestyle experience. With features such as user profiles, fashion blogs, trend updates, and a responsive shopping module, the platform ensures easy navigation and engagement. The efficient order management further improves customer satisfaction. Moving forward, continuous improvements based on user feedback, expansion of product offerings, and enhanced marketing strategies will help the platform grow and attract a wider audience. This project lays a strong foundation for future developments in the digital fashion and lifestyle industry.