A Project Report on

eFuel: An Online Fuel Home Delivery Platform

*is submitted in partial fulfillment of the requirement for the award of the Degree of*

***BACHELOR OF TECHNOLOGY***

*to*

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTHAPURAMU**

### by

**Y SAI BHAVESH REDDY (20AT1A05C1) K KIRAN KUMAR (21AT5A0509)**

### G RAGHAVENDRA (20AT1A05B0)

**Under the Guidance of**

**Dr. S PREM KUMAR M.Tech.,Ph.D**

### Assistant Professor



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

## G.PULLAIAH COLLEGE OF ENGINEERING AND TECHNOLOGY

### (Autonomous)

Approved by AICTE | NAAC Accreditation with ‘A’ Grade

Accredited by NBA (CSE, ECE & EEE) | Permanently Affiliated to JNTUA Nandikotkur Road, Venkayapalli (V), Kurnool - 518452, Andhra Pradesh

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

## G. PULLAIAH COLLEGE OF ENGINEERING AND TECHNOLOGY

### (Autonomous)

Approved by AICTE | NAAC Accreditation with ‘A’ Grade

Accredited by NBA (CSE, ECE & EEE) | Permanently Affiliated to JNTUA Nandikotkur Road, Venkayapalli (V), Kurnool - 518452, Andhra Pradesh



**CERTIFICATE**

This is to certify that the project report entitled **“eFuel: An Online Fuel Home Delivery Platform”** being submitted by **Y Sai Bhavesh Reddy (20AT1A05C1) K Kiran Kumar (21AT5A0509) G Raghavendra (20AT1A05B0)** in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering of G. Pullaiah College of Engineering and Technology, Kurnool is a record of bonafide work carried out by them under my guidance and supervision. The results embodied in this project report have not been submitted to any other university or institute for the award of any Degree or Diploma.

Internal Guide Head of the Department

**Dr. S PREM KUMAR M.Tech., Ph.D Dr. M. SRI LAKSHMI**

Professor Associate Professor

Date of Viva-Voce

INTERNAL EXAMINER EXTERNAL EXAMINER

# ACKNOWLEDGEMENT

We thank our project supervisor, **Dr. S PREM KUMAR M.Tech.,Ph.D** for his guidance, valuable suggestions, and support in the completion of the project.

We would like to express our deep sense of gratitude and our sincere thanks to HOD

**Mrs M. SRI LAKSHMI M.Tech.,(Ph.D).** Department of Computer Science and Engineering,

G. Pullaiah College of Engineering and Technology, Kurnool for providing the necessary facilities and encouragement towards the project work.

We owe indebtedness to our principal **Dr. C. SREENIVASA RAO M.Tech., Ph.D.**

G. Pullaiah College of Engineering and Technology, Kurnool for providing us the required facilities.

We are extremely grateful to Chairman **Mr. G. V. M. MOHAN KUMAR** of

G. Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh for their good blessings. We gratefully acknowledge and express our thanks to the teaching and non-teaching staff of the CSE Department.

## Project Associates

### Y SAI BHAVESH REDDY (20AT1A05C1) K KIRAN KUMAR (21AT5A0509)

**G RAGHAVENDRA (20AT1A05B0)**

# ABSTRACT

This paper presents about eFuel which is a comprehensive full-stack web application designed to revolutionize the way people access fuel for their homes. This project aims to provide a convenient and efficient solution for ordering fuel online and having it delivered directly to their doorstep. The platform features user-friendly sign up and sign in processes, allowing customers to create accounts and manage their fuel orders with ease. eFuel also determine the user's fuel choice, streamlining the ordering process further. Users can specify their fuel requirements, including the type of fuel (e.g., petrol, diesel), the preferred fuel company, and the desired number of liters. The application integrates real-time data from fuel suppliers to provide up-to-date pricing information and availability, ensuring that customers can make informed choices.

**Keywords**: Fuel delivery, Full-stack web application, User-friendly, Convenience.

## CONTENTS

### Abstract

**CHAPTER 1 INTRODUCTION**

1.1 Introduction

### CHAPTER 2 LITERATURE REVIEW

2.1 Literature survey

### CHAPTER 3 PROPOSED METHOD AND EXISTING METHOD

* 1. Existing System
  2. Proposed System
  3. Modules
  4. Methodology

### CHAPTER 4 SYSTEM REQUIREMENTS

* 1. Hardware Requirements
  2. Software Requirements
  3. Libraries
     1. JDBC
     2. Java Servlet API
     3. Apache Tomcat
     4. Java EE
     5. HTML AND CSS
     6. BOOTSTRAP

### CHAPTER 5 APPLICATION DETAILS

* 1. Details
  2. Advantages of eFuel Website

### CHAPTER 6 FRONTEND TECHNOLOGIES

* 1. HTML
  2. CSS(BootStrap)
  3. JAVASCRIPT

### CHAPTER 7 BACKEND TECHNOLOGIES

* 1. Java Servlet
  2. MySQL
  3. Apache Tomcat Server

## CONTENTS

### CHAPTER 8 SOURCE CODE AND IMPLEMENTATION

* 1. Source Code
  2. Implementation

### CHAPTER 9 CONCLUSION

9.1 Conclusion

**BIBLIOGRAPHY**

**1.1 Introduction:**

# CHAPTER 1 INTRODUCTION

In our digitally-driven world, convenience and efficiency have become paramount in nearly every aspect of daily life, including accessing essential resources like fuel for our homes. Introducing eFuel, a cutting-edge full-stack web application poised to transform the way individuals procure fuel with unprecedented ease and convenience. With a primary goal of streamlining the process of ordering fuel online and delivering it directly to customers' doorsteps, eFuel represents a significant leap forward in fuel access technology. At the core of eFuel lies its user-centric design, meticulously crafted to offer a seamless experience from sign- up to delivery. With intuitive sign-up and sign-in processes, users can effortlessly create accounts and manage their fuel orders with unparalleled convenience. Leveraging state-of-the- art geolocation services, eFuel further enhances user experience by intuitively determining their fuel preferences, eliminating unnecessary steps and minimizing time spent on the platform.One of eFuel's standout features is its customizable order system, empowering users to tailor their fuel procurement to their exact needs. Whether it's specifying the type of fuel – be it petrol, diesel – selecting preferred fuel providers, or indicating the desired quantity in liters, eFuel puts control firmly in the hands of the consumer. Moreover, by seamlessly integrating real-time data from a network of reputable fuel suppliers, the platform ensures that users have access to the most current pricing information and availability, empowering them to make informed decisions with confidence.

In summary, eFuel represents a paradigm shift in the way individual access and manage their home fuel needs. By marrying cutting-edge technology with user-centric design principles, eFuel offers a transformative solution that not only simplifies the fuel procurement process but also enhances transparency, efficiency, and overall customer satisfaction. Join us as we embark on this journey to revolutionize home fuel access with eFuel – where convenience meets innovation."

**2.1 Literature survey:**

# CHAPTER 2

**LITERATURE SURVEY**

This paper presents the various methods that can be used for ordering fuel online. In the realm of e-commerce, studies on user interface design emphasize optimizing customer experience in fuel ordering systems, particularly focusing on cart management and order submission processes. Administrative control interfaces are also under scrutiny, with research comparing approaches to system management, including user account management and product inventory control. Security measures in e-commerce platforms are a significant focus, examining authentication protocols and data encryption techniques to safeguard user data and transactions in fuel ordering systems. Customer relationship management strategies are explored to retain and engage users, delving into personalized marketing tactics and feedback mechanisms to drive loyalty. Inventory management techniques are studied to ensure timely fulfillment of customer orders, analyzing inventory tracking systems and supply chain logistics. Data analytics play a crucial role, with research focusing on leveraging customer insights for business growth in fuel ordering platforms through predictive modeling and business intelligence tools. Ethical considerations in e-commerce are addressed, emphasizing fairness and transparency in data privacy and pricing practices within fuel procurement processes. Finally, future trends in e-commerce technology are explored, forecasting innovations such as AI and IoT to further optimize fuel ordering systems.

# CHAPTER 3

**EXISTING SYSTEM AND PROPOSED SYSTEM**

## Existing system:

**Limited Accessibility and Convenience:** Many existing systems rely on users physically visiting traditional fuel stations to refill their tanks. This process often involves waiting in line and dedicating time and effort to refueling, especially during peak hours or inclement weather conditions. Such limitations can inconvenience users and disrupt their daily routines, highlighting the need for more convenient alternatives like eFuel's online ordering and doorstep delivery service.

**Fragmented Information and Lack of Transparency**: Some current platforms may struggle to integrate seamlessly with various fuel suppliers, leading to fragmented information regarding pricing, availability, and delivery options. This fragmentation can create confusion for users, making it challenging for them to compare options and make informed decisions. By consolidating real-time data from multiple suppliers, eFuel addresses this issue, providing users with transparent and up-to-date information to facilitate confident purchasing choices.

**Limited Customization and Personalization:** Existing platforms often lack the level of customization and personalization that eFuel offers. Users may be restricted in specifying detailed requirements such as fuel type, preferred supplier, and desired quantity, resulting in a less tailored experience. By empowering users with the ability to personalize their orders according to their specific preferences and needs, eFuel enhances user satisfaction and loyalty, setting it apart from conventional fuel ordering systems.

## Proposed system:

The proposed system, eFuel, is a comprehensive online platform designed to revolutionize the way individuals access and procure fuel for their homes. It aims to provide a seamless and convenient solution for ordering fuel online and having it delivered directly to the user's doorstep.

**Online Ordering and Delivery:** The core functionality of eFuel is its online ordering system, allowing users to browse fuel options, add products to their cart, and submit orders for delivery. This eliminates the need for users to physically visit traditional fuel stations, offering a more convenient and time- saving alternative.

**Seamless Integration with Fuel Suppliers:** eFuel integrates seamlessly with a network of reputable fuel suppliers, ensuring users have access to real-time information on pricing, availability, and delivery options. By consolidating data from multiple suppliers into a single platform, eFuel provides users with a streamlined and transparent purchasing experience.

**Customization and Personalization:** One of the key advantages of eFuel is its emphasis on customization and personalization. Users have the flexibility to specify detailed requirements for their fuel orders, including fuel type (e.g., petrol, diesel), preferred supplier, and desired quantity. This level of customization enhances user satisfaction and allows for a more tailored experience.

**User-Friendly Interface:** eFuel features a user-friendly interface, designed to be intuitive and easy to navigate. Whether users are browsing fuel options, managing their orders, or updating their account information, the interface is optimized for a seamless user experience.

**Account Management:** The system includes robust account management functionality, allowing users to create accounts, log in securely, and manage their profiles. Users can update their personal information, view order history, and track the status of their current orders.

**Admin Dashboard**: eFuel includes an admin dashboard that provides administrators with comprehensive control over the platform's operations. Admins can manage user accounts, view order details, update product information, and communicate with users as needed.

### Conclusion:

The proposed system, eFuel, offers a user-centric and technologically advanced solution for ordering fuel online. With its seamless integration with fuel suppliers, emphasis on customization, and user- friendly interface, eFuel aims to revolutionize the way individuals access and procure fuel for their homes, providing unparalleled convenience and efficiency.

## Modules:

### User Management:

Handles user accounts, registration, and login functionalities for a straight forward user experience.

### Order Processing:

Manages the entire fuel ordering process, from placement to delivery, ensuring simplicity and efficiency.

### Supplier Integration:

Integrate seamlessly with fuel suppliers to provide users with real-time pricing and availability information, facilitating well-informed, decision-making.

## METHODOLOGY:

### Agile Methodology:

Begin by understanding the project requirements and user needs. Define user stories to capture specific features and functionalities required for the eFuel platform, such as user registration and order processing. Divide the project into smaller iterations known as sprints. Plan the features and tasks to be implemented in each sprint based on priority and complexity. Implement features according to the user stories outlined in the sprint planning. This involves coding the frontend (HTML,CSS(Bootstrap), JavaScript) and backend (server-side logic using a programming language Jsp, Java Servlet, Tomcat Server), as well as setting up the database (SQL).

# CHAPTER 4 SYSTEM REQUIREMENTS

## HARDWARE REQUIREMENTS:

|  |  |  |
| --- | --- | --- |
| • | Processor | : i4 Processor |
| • | Hard Disk | : 100 GB |
| • | Ram | : 4GB |

* 1. **SOFTWARE REQUIREMENTS:**
     + Operating System : Windows 11
     + Coding Language : Java(Jsp and Java Servlet)
     + Front End : HTML,CSS, JS
     + Back End : Apache Tomcat server, My SQL

## LIBRARIES

* + - JDBC
    - Java Servlet API
    - Apache Tomcat
    - Java EE
    - HTML and CSS
    - Bootstrap

**JDBC (Java Database Connectivity):** JDBC is used to interact with the database and execute SQL queries. It provides a standard Java API for accessing databases.

**Java Servlet API:** The project likely uses Java Servlets for handling HTTP requests and generating dynamic web content. Servlets are Java classes that extend the functionality of web servers to process client requests and respond with dynamic content.

**Apache Tomcat (or a similar servlet container):** Apache Tomcat is a popular servlet container that allows Java web applications to run on a web server. It provides an environment for executing servlets and managing web applications.

**Java EE (Enterprise Edition):** The project may utilize other components of the Java EE platform, such as JSP (JavaServer Pages) for generating dynamic web pages, and JNDI (Java Naming and Directory Interface) for accessing resources like databases.

**HTML and CSS:** HTML is used for structuring web pages, while CSS is used for styling and formatting the appearance of the web pages. These are fundamental technologies for building web applications.

**JavaScript:** JavaScript may be used for client-side scripting to enhance the interactivity and functionality of the web pages. It can be used for tasks such as form validation, handling events, and making asynchronous requests to the server.

**Bootstrap (from maxcdn.bootstrapcdn.com):** Bootstrap is a popular front-end framework for building responsive and mobile-friendly web applications. It provides pre-designed CSS and JavaScript components that help streamline the development process. In the provided code, resources from maxcdn.bootstrapcdn.com are loaded, indicating the use of Bootstrap for styling and layout.

These are the main libraries and technologies inferred from the provided code snippet. Depending on the specific requirements and architecture of the project, additional libraries or frameworks may be used.

* 1. **Details**

# CHAPTER 5

**APPLICATION DETAILS**

**User Registration:** Users can register for an account on the e-commerce platform by providing their personal details such as name, email address, and password.

**User Authentication:** Registered users can log in to their accounts using their email address and password. Authentication mechanisms ensure that only authorized users can access their accounts and perform actions.

**Product Browsing:** Users can browse through a catalog of products available on the platform. Products are displayed with details such as title, description, price, and product image.

**Product Search:** Users can search for specific products using keywords or filters such as category, price range, or brand. The search functionality allows users to quickly find products of interest.

**Product Details:** Clicking on a product displays detailed information about the selected item, including its description, price, availability, and related images.

**Adding Products to Cart:** Users can add products to their shopping cart by specifying the desired quantity. The cart keeps track of selected items, allowing users to review and modify their selections before proceeding to checkout

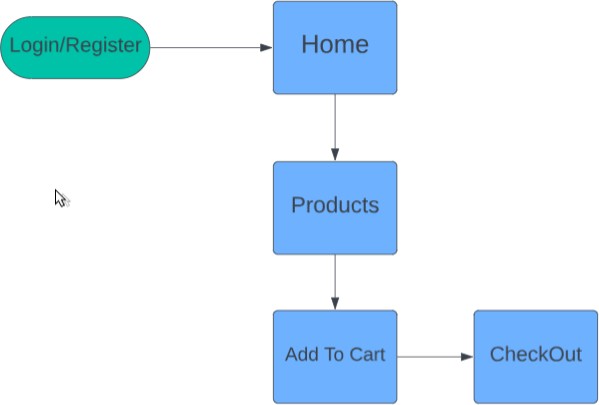


fig 5.1 Working of eFuel Website

**Cart Management:** Users can view the contents of their shopping cart, update quantities, remove items, or clear the entire cart. Cart management functionality ensures a seamless shopping experience.

**Order Placement:** Once users are satisfied with their selections, they can proceed to checkout to place their orders. During checkout, users provide shipping and payment information for order fulfillment.

**Order Confirmation**: Upon successful order placement, users receive confirmation messages with details such as order number, total cost, and estimated delivery date. Confirmation emails may also be sent to users' registered email addresses.

**Order History:** Users can view their order history, including details of past purchases such as order dates, products ordered, quantities, and total costs. Order history functionality allows users to track their purchases and reorder items if needed.

**Account Management:** Users can update their account information, including personal details, shipping addresses, and payment methods. Account management features ensure that users can maintain accurate and up-to-date profiles.

**Admin Dashboard:** Administrators have access to an admin dashboard where they can manage various aspects of the e-commerce platform. Admin functionalities include managing products, categories, users, orders, and website settings.

**Product Management:** Administrators can add new products, update existing product details, and delete products from the catalog. Product management tools allow administrators to maintain an up-to- date and organized product inventory.

**Category Management:** Administrators can create, edit, and delete product categories to organize the product catalog. Category management ensures that products are logically grouped and easy to navigate for users.

**User Management**: Administrators can manage user accounts, including creating new accounts, updating user details, resetting passwords, and deactivating or deleting accounts as needed. User management functionality ensures proper user administration and security.

**Order Management:** Administrators can view and manage orders placed by users, including order status, shipping details, payment information, and order fulfillment. Order management tools facilitate efficient order processing and tracking.

## Advantages of eFuel Website:

**Convenience:** Users can order fuel from the comfort of their homes without visiting physical fuel stations.

**Time-saving:** Ordering fuel online eliminates the need to wait in line at fuel stations, saving users valuable time.

**Accessibility:** The platform is accessible 24/7, allowing users to place orders at their convenience.

**Transparency:** Users have access to real-time pricing and availability information from multiple fuel suppliers.

**Customization:** Users can specify detailed requirements for their fuel orders, such as quantity and preferred supplier.

**User-friendly interface:** The platform's interface is intuitive and easy to navigate, enhancing user experience.

* 1. **HTML**

# CHAPTER 6

**FRONTEND TECHNOLOGIES**

HTML (Hypertext Markup Language) is the standard markup language used to create and design web pages. It provides the structure and content of a webpage, defining elements such as headings, paragraphs, links, images, tables, and forms. Here's a brief overview of HTML: **Elements**: HTML documents are composed of HTML elements, which are enclosed in tags and define the structure and content of the webpage. Elements consist of an opening tag, content, and a closing tag. For example:

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

**Attributes**: HTML elements can have attributes that provide additional information about the element, such as its appearance or behavior. Attributes are specified within the opening tag of an element. For example:

<img src="image.jpg" alt="Image Description">

<a href="https://[www.example.com"](http://www.example.com/) target="\_blank">Visit Example</a>

**Document Structure**: HTML documents have a hierarchical structure defined by nested elements. The <html> element serves as the root of the document, containing two main **sections**: <head> and <body>. The <head> section contains meta-information about the document, such as the title and links to stylesheets, while the <body> section contains the visible content of the webpage.

**Semantic Elements**: HTML5 introduced semantic elements that provide meaning to the content of a webpage, making it more accessible to both humans and machines. Examples of semantic elements include <header>, <nav>, <main>, <section>, <article>, <footer>, etc. **Links and Navigation**: HTML allows you to create hyperlinks using the <a> (anchor) element, which allows users to navigate between different web pages or sections within the same page. Links can point to other web pages, specific sections within the same page (using anchor links), or external resources like images and documents.

**Images and Multimedia**: HTML provides elements such as <img> for embedding images,

<audio> for embedding audio files, and <video> for embedding video files. These elements allow you to include multimedia content within your web pages.

## CSS(BOOTSTRAP)

CSS stands for Cascading Style Sheets, and it is a style sheet language used for describing the presentation of a document written in a markup language like HTML.

CSS describes how elements should be displayed on a webpage, including aspects such as layout, colors, fonts, and spacing.

### Introduction to Bootstrap:

Bootstrap is a free and open-source front-end framework developed by Twitter.

It provides pre-designed HTML and CSS-based design templates for typography, forms, buttons, navigation, and other interface components, as well as optional JavaScript extensions.

Bootstrap is designed to be mobile-first and responsive, meaning that it automatically adjusts the layout and appearance of web pages to fit different screen sizes and devices.

Key Features of Bootstrap:

Grid System: Bootstrap utilizes a responsive, mobile-first grid system based on a 12-column layout. This grid system allows developers to create responsive layouts that adapt to various screen sizes and resolutions.

CSS Components: Bootstrap provides a wide range of pre-designed CSS components, such as buttons, forms, navigation bars, dropdowns, alerts, badges, and more. These components can be easily integrated into web pages to enhance their functionality and appearance.

Responsive Design: Bootstrap's responsive design features enable web pages to adapt and scale seamlessly across different devices and screen sizes, including desktops, laptops, tablets, and smartphones. This ensures a consistent user experience across all devices.

Customizable Styles: While Bootstrap comes with predefined styles and components, it also offers extensive customization options. Developers can modify and extend Bootstrap's styles using custom CSS or Sass variables to match the design requirements of their projects.

Browser Compatibility: Bootstrap is designed to be compatible with modern web browsers, ensuring consistent rendering and functionality across different browsers and browser versions. It includes built- in CSS normalization and support for vendor prefixes to address cross-browser inconsistencies.

Community Support: Bootstrap has a large and active community of developers, designers, and contributors who regularly contribute to its development, provide support, and share resources such as themes, templates, and plugins. This community-driven approach fosters collaboration and innovation within the Bootstrap ecosystem.

### How Bootstrap Works:

Integration: To use Bootstrap in a project, developers can include the Bootstrap CSS and JavaScript files in their HTML documents. Bootstrap can be integrated into projects in various ways, including

downloading the Bootstrap files and linking them locally, or using Content Delivery Networks (CDNs) to link to hosted Bootstrap files.

HTML Structure: Bootstrap components are implemented using HTML markup with predefined CSS classes. Developers can add these classes to HTML elements to apply Bootstrap styles and functionality to them. For example, adding the "btn" class to a button element will style it as a Bootstrap button.

Responsive Grid: Bootstrap's grid system allows developers to create responsive layouts by dividing the page into rows and columns. Developers can specify the size of each column and how it should behave on different screen sizes using Bootstrap's predefined grid classes (e.g., col-md-6 for a column that spans half the width on medium-sized screens).

Component Customization: Bootstrap components can be customized using CSS or Sass variables to modify colors, fonts, sizes, spacing, and other visual properties. Developers can override Bootstrap's default styles or create their own custom styles to match the design requirements of their projects.

JavaScript Plugins: Bootstrap includes optional JavaScript plugins for adding interactive functionality to components such as dropdowns, modals, carousels, tooltips, and more. Developers can enable these plugins by including the Bootstrap JavaScript file in their projects and initializing the plugins using JavaScript code.

### Advantages of Using Bootstrap:

Rapid Development: Bootstrap provides pre-designed CSS and HTML components that can be easily integrated into projects, speeding up the development process and reducing the need for custom styling and scripting.

Consistency: Bootstrap ensures consistency in design and layout across web pages, making it easier for users to navigate and interact with websites and web applications.

Responsive Design: Bootstrap's responsive grid system and components enable developers to create mobile-first, responsive designs that adapt to different screen sizes and devices, enhancing usability and accessibility.

Cross-browser Compatibility: Bootstrap is designed to be compatible with modern web browsers, ensuring consistent rendering and functionality across different browser platforms and versions.

## JAVASCRIPT

JavaScript is a high-level programming language primarily used for adding interactivity and dynamic behavior to web pages. It is an essential component of web development, allowing developers to create interactive features, handle user input, manipulate HTML and CSS, and communicate with web servers. Here's a brief overview of JavaScript:

**Client-side Scripting**: JavaScript is primarily used for client-side scripting, meaning it runs in the user's web browser rather than on a web server. This allows JavaScript code to interact with the Document Object Model (DOM) of a webpage, enabling dynamic manipulation of page content, structure, and styling.

**Event-driven Programming**: JavaScript follows an event-driven programming model, where code execution is triggered by user actions (e.g., clicks, mouse movements, keyboard input) or by other events (e.g., page load, form submission, timer expiration). Event handlers are used to define JavaScript functions that respond to these events.

**Data Types and Variables**: JavaScript supports various data types, including numbers, strings, booleans, arrays, objects, functions, and more. Variables are used to store and manipulate data in JavaScript, with dynamic typing allowing variables to change their data type during runtime. **Control Flow and Loops**: JavaScript provides control flow statements such as if-else statements, switch-case statements, and loops (e.g., for loops, while loops) for implementing conditional logic and repetitive tasks. These control flow structures allow developers to control the flow of program execution based on conditions and iteration.

**Asynchronous Programming**: JavaScript supports asynchronous programming through callbacks, promises, and async/await syntax. Asynchronous operations, such as fetching data from a web server or performing file I/O, allow JavaScript applications to remain responsive and handle long-running tasks without blocking the main execution thread.

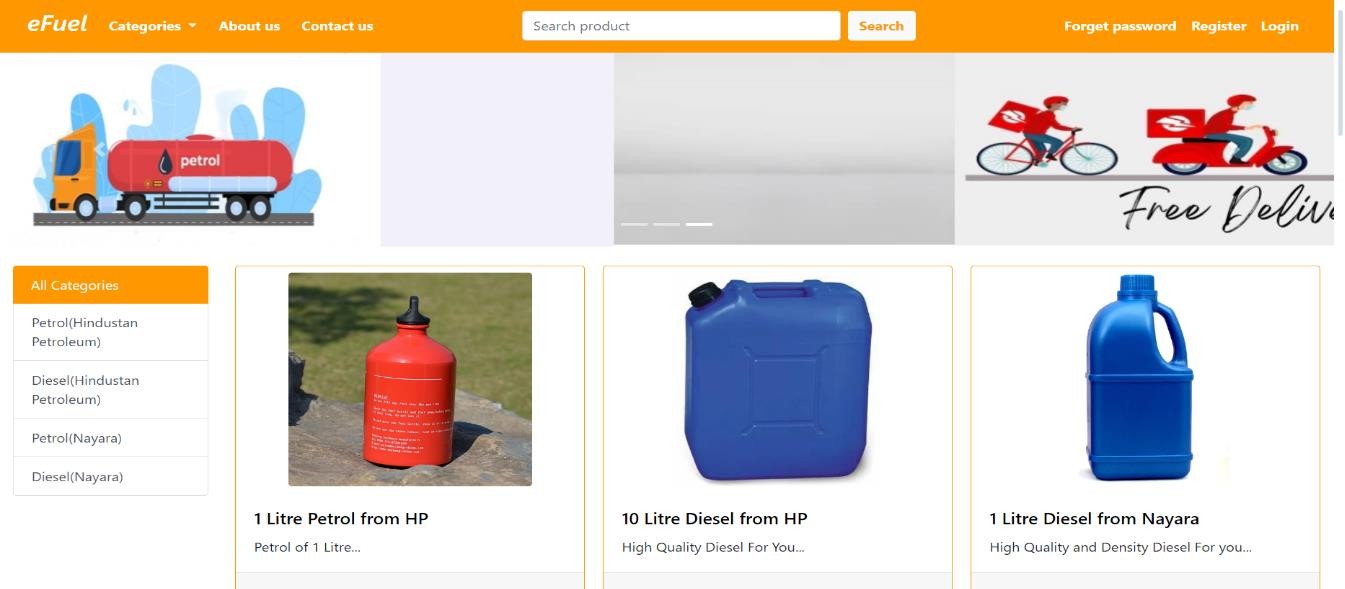


fig 6.4 home page

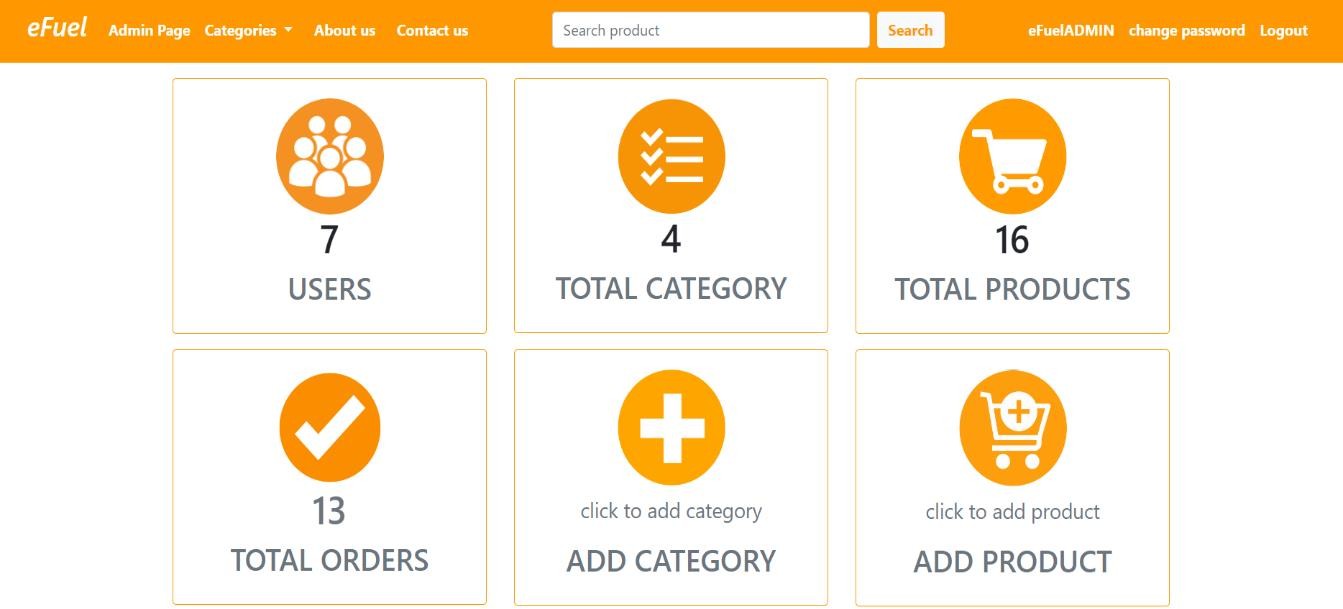
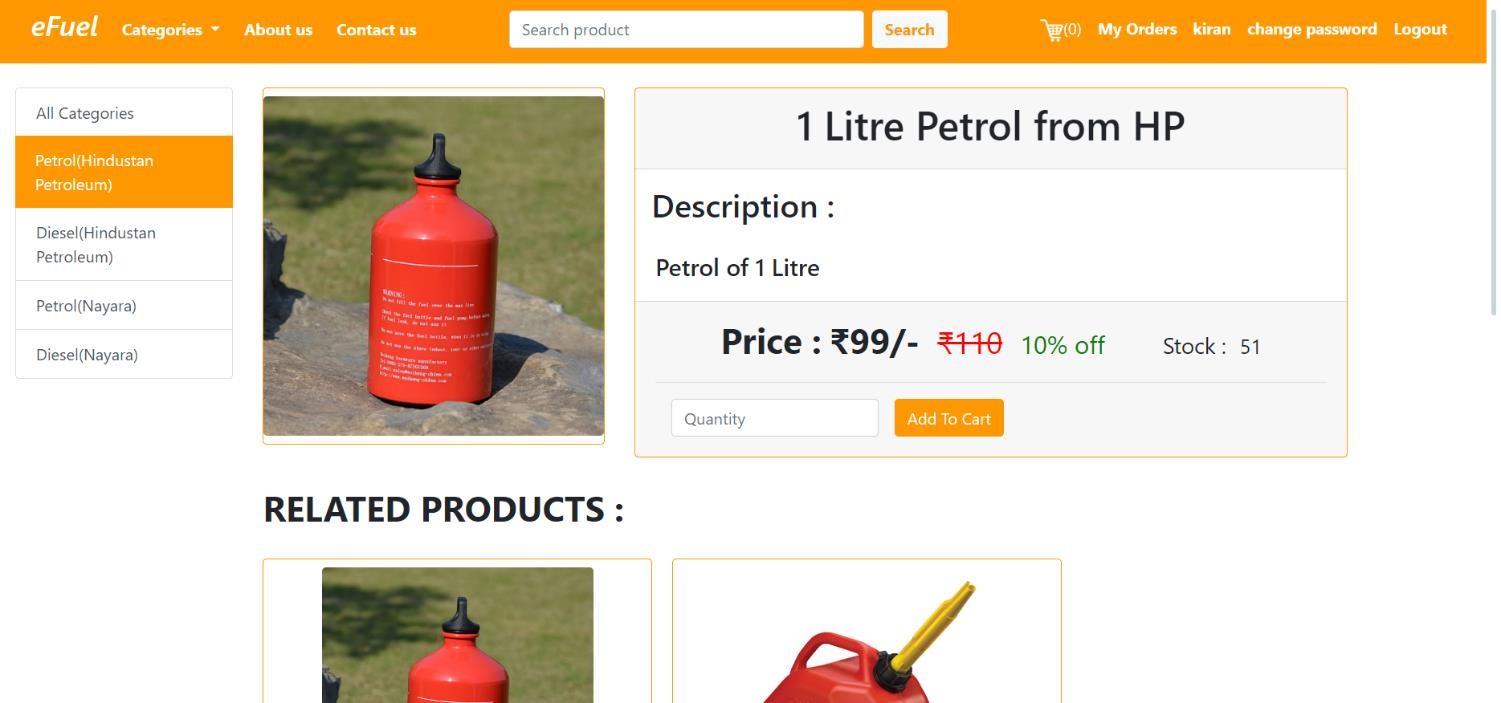
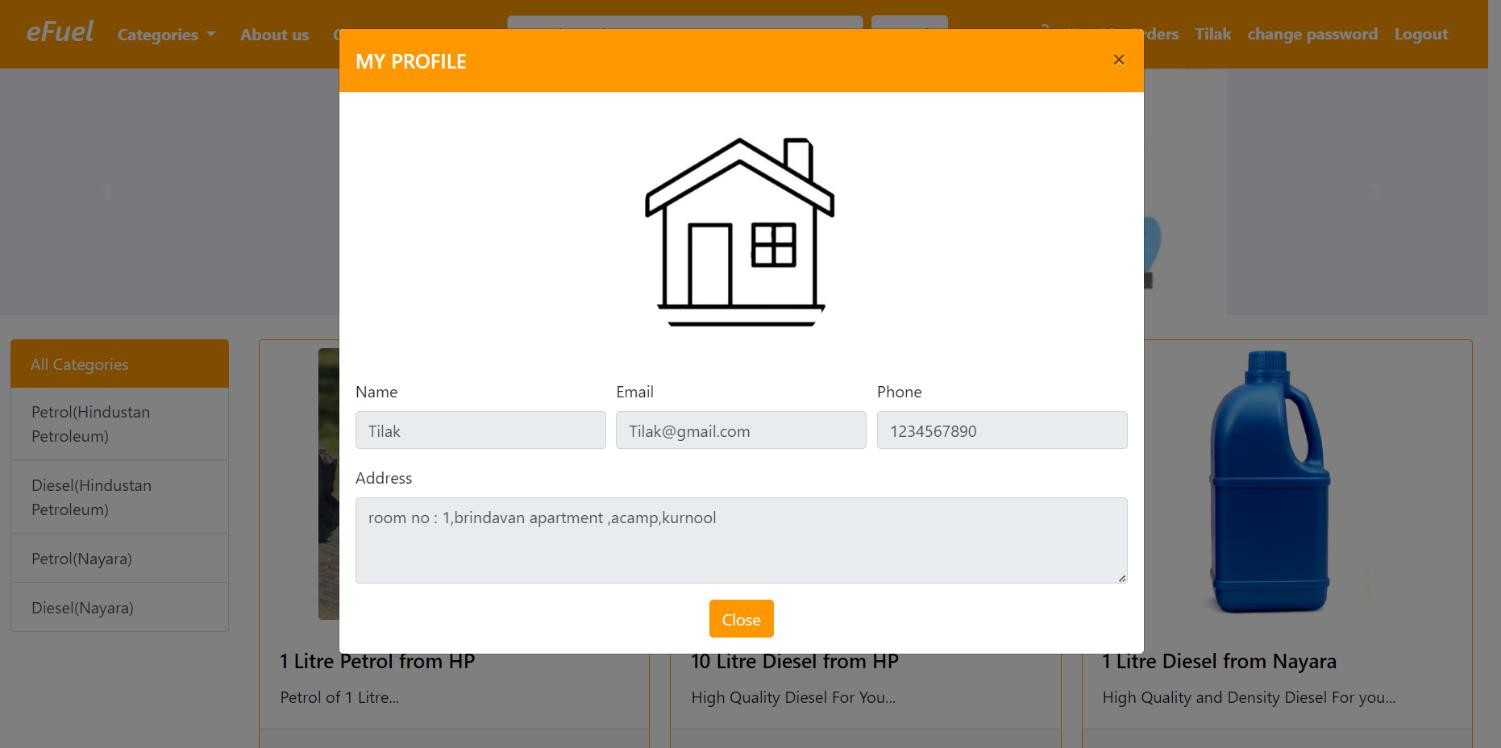


fig 6.5 admin page



* 1. product selecting page



* 1. customer profile page

# CHAPTER 7

**BACKEND TECHNOLOGIES**

## JAVA SERVLET

Java servlets are server-side components written in Java that extend the functionality of web servers to handle HTTP requests and generate dynamic responses. They follow a lifecycle model, initialized when loaded into memory and destroyed when no longer needed. Servlets can handle various types of HTTP requests, manage user sessions, access databases, enforce security measures, and generate dynamic content based on request parameters and application logic. They are lightweight, efficient, and portable, running on any Java-enabled web server. Servlets provide a flexible and scalable foundation for building modern web applications, offering features such as session management, database access, and security enforcement to create interactive and robust web experiences.

## MYSQL

MySQL is an open-source relational database management system renowned for its reliability, flexibility, and scalability. As a relational database, MySQL stores data in tables with rows and columns, making it easy to organize and query structured information using SQL commands. It ensures data integrity through ACID compliance, guaranteeing transaction reliability. MySQL's scalability features, including replication and clustering, enable it to handle growing workloads and data volumes. With its high performance and robust feature set, MySQL is a popular choice for powering a wide range of applications, from small-scale websites to enterprise-level systems, across various industries.

## Apache Tomcat Server

Apache Tomcat, often referred to simply as Tomcat, is an open-source web server and servlet container developed by the Apache Software Foundation. It serves as a Java-based application server for running Java Servlets, JavaServer Pages (JSP), and other Java-based web applications. Tomcat implements the Java Servlet and JavaServer Pages specifications, providing a lightweight and efficient environment for deploying and managing dynamic web content. It is widely used for hosting Java web applications due to its ease of use, robustness, and scalability. Tomcat's modular architecture allows for easy configuration and extension, enabling developers to customize and optimize the server to meet their specific requirements. With its reliability, performance, and extensive community support, Apache Tomcat remains a popular choice for hosting Java-based web applications across various industries.

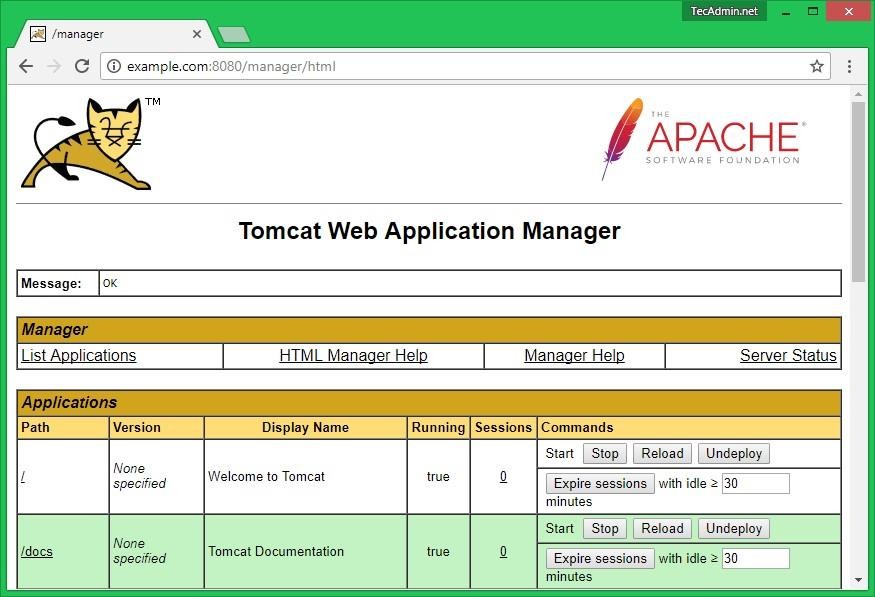


Fig 7.3 Tomcat environment

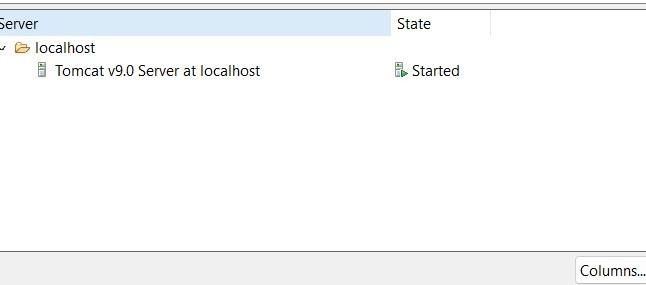


Fig 7.4 Initialisation of tomcat

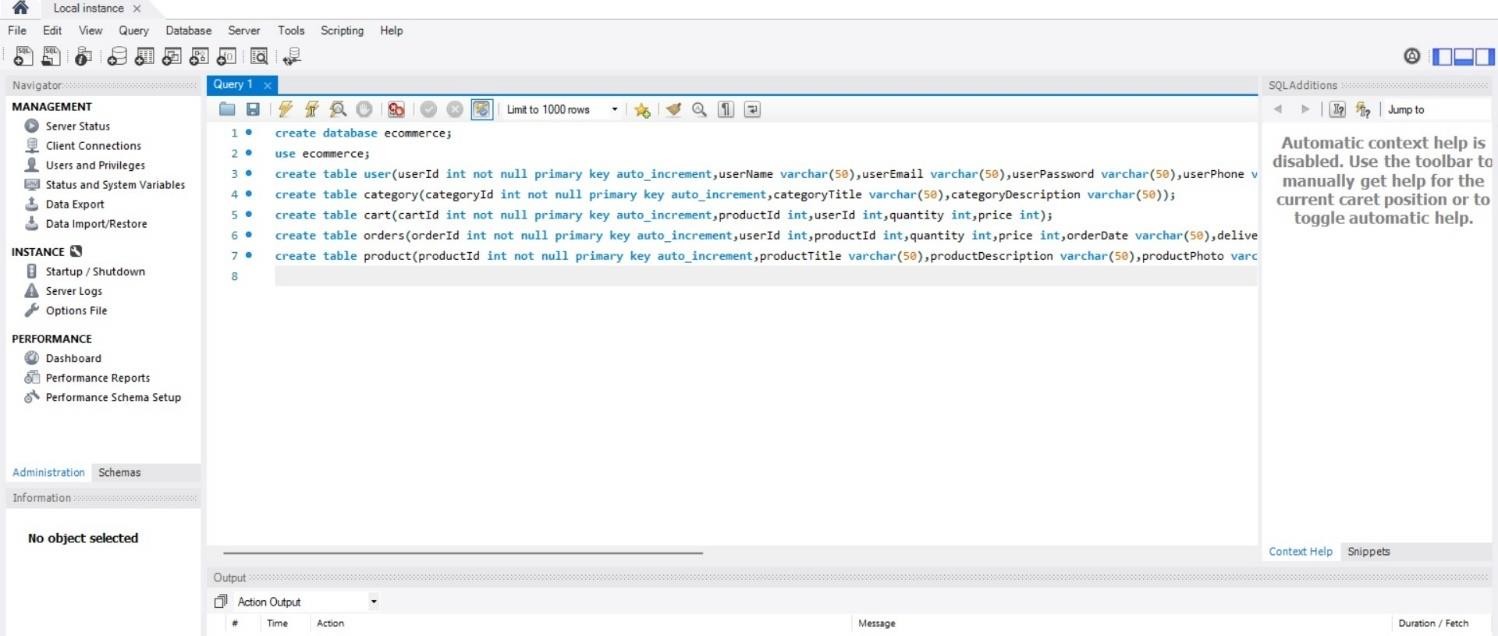


Fig 7.5 database

# CHAPTER 8

**SOURCE CODE AND IMPLEMENTATION**

## Source code

page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ page import =*"com.ecommerce.pojo.\*"*%>

<%@ page import =*"com.ecommerce.dao.\*"*%>

<%@ page import =*"java.sql.\*"*%>

<%@ page import =*"java.util.\*"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Home- eFuel</title>

<%@ include file=*"components/common\_cs\_js.jsp"*%>

</head>

<body>

<%@ include file=*"components/navbar.jsp"*%>

<%

String searchResult=(String)request.getAttribute("enteredText"); String cat=request.getParameter("category");

ProductDaoImp pdao=**new** ProductDaoImp(); CategoryDaoImp cdao1=**new** CategoryDaoImp(); List<Product> l1=**null**;

List<Category> cl1=cdao1.getAllCategory();

**if**(cat==**null**)

{

cat="0";

**if**(cat.trim().equals("0"))

{

l1 =pdao.getAllProduct();

}

}

**else if**(cat.equals("0"))

{

l1 =pdao.getAllProduct();

}

**else**

{

**int** categoryId=Integer.parseInt(cat.trim()); l1=pdao.getProductByCategory(categoryId);

}

%>

<div id=*"carouselExampleIndicators"* class=*"carousel slide"* data- ride=*"carousel"*>

<ol class=*"carousel-indicators"*>

<li data-target=*"#carouselExampleIndicators"* data-slide-to=*"0"* class=*"active"*></li>

<li data-target=*"#carouselExampleIndicators"* data-slide-to=*"1"*></li>

<li data-target=*"#carouselExampleIndicators"* data-slide-to=*"2"*></li>

</ol>

<div class=*"carousel-inner"* style="width:*100%*">

<div class=*"carousel-item active"*>

<img class=*"d-block w-100"* src=*"images/ca1.jpg"* alt=*"First slide"*>

</div>

<div class=*"carousel-item"*>

<img class=*"d-block w-100"* src=*"images/ca2.jpg"* alt=*"Second slide"*>

</div>

<div class=*"carousel-item"*>

<img class=*"d-block w-100"* src=*"images/ca3.jpg"* alt=*"Third slide"*>

</div>

</div>

<a class=*"carousel-control-prev"* href=*"#carouselExampleIndicators"* role=*"button"* data-slide=*"prev"*>

<span class=*"carousel-control-prev-icon"* aria-hidden=*"true"*></span>

<span class=*"sr-only"*>Previous</span>

</a>

<a class=*"carousel-control-next"* href=*"#carouselExampleIndicators"* role=*"button"* data-slide=*"next"*>

<span class=*"carousel-control-next-icon"* aria-hidden=*"true"*></span>

<span class=*"sr-only"*>Next</span>

</a>

</div>

<div class=*"text-center"*><%@ include file=*"components/message.jsp"*%></div>

<div class=*"container-fluid"*>

<!-- show categories -->

<div class=*"row"*>

<div class=*"col-md-2 "* >

<div class=*"list-group mt-4"*>

<%

**int** catId=Integer.parseInt(cat);

%>

<a href=*"index.jsp?category=0"* class=*"list-group-item list- group-item-action* <%**if**(catId==0){ %> *active custom-bg*<%} %>*"* <%**if**(catId==0){ %> style="border:*#ff9800*"<%} %> >All Categories</a>

<%

**for**(Category c: cl1)

{

%>

<a href=*"index.jsp?category=*<%=c.getCategoryId() %>*"* class=*"list-group-item list-group-item-action* <%**if**(catId==c.getCategoryId()){ %> *active custom-bg*<%} %>*"* <%**if**(catId==c.getCategoryId()){ %> style="border:*#ff9800*"<%} %>><%=c.getCategoryTitle() %></a>

<%

}

%>

</div>

</div>

<!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->

<!-- show products -->

<%

**if**(searchResult==**null**)

{

%>

<div class=*"col-md-10 "*>

<div class=*"row mt-4"*>

<div class=*"col-md-12 admin"*>

<div class=*"card-columns hover"*>

<%

String stock="Out Of Stock!!!";

**for**(Product p:l1)

{

%>

<div class=*"card"*>

<img src=*"productImages/*<%=p.getProductPhoto() %>*"* style="max-height:*270px*;max-width:*100%*;width:*auto*;" class=*"card-img-top rounded mx-auto d-block m-2"* alt=*"img"*>

<div class=*"card-body"*>

<h5 class=*"card-title"*><a href=*"product.jsp?productId=*<%=p.getProductId()%>*"* style="text-decoration: *none*;color:*black*;"> <%=p.getProductTitle() %></a></h5>

<p class=*"card- text"*><%=Helper.get10Words(p.getProductDescription()) %></p>

</div>

<div class=*"card-footer text-center"*>

<p style="font-size:*25px*"><span class=*"ml- 2"*><b>&#8377;<%=Helper.getProductSellingPrice(p.getProductPrice(), p.getProductDiscount()) %>/-</b></span>

<span class=*"ml-2"* style="font- size:*20px*;color:*red*"><s>&#8377;<%=p.getProductPrice()%></s></span>

<span class=*"ml-2"* style="font- size:*20px*;color:*green*"><%=p.getProductDiscount() %>&#37 off</span>

</p>

<span class=*"ml-2"* style="font-

size:*20px*;">Stock :</span>

<span class=*"ml-1"* style="font-

size:*20px*"><%**if**(p.getProductQuantity()<1){ %><span style="color:*red*;"><b><%=stock%></b></span><%} **else**{

%><%=p.getProductQuantity()%><% } %></span>

</div>

</div>

<%

}

%>

</div>

</div>

</div>

</div>

<%

}

**else**

{

List<Product> listOfProductFromSearch=**new** ProductDaoImp().getSearchedProduct(searchResult);

%>

<div class=*"col-md-10"*>

<div class=*"row mt-4"*>

<div class=*"col-md-12"*>

<div class=*"card-columns"*>

<%

String stock="Out Of Stock!!!";

**for**(Product p:listOfProductFromSearch)

{

%>

<div class=*"card p-2"* >

<img src=*"productImages/*<%=p.getProductPhoto() %>*"* style="max-height:*270px*;max-width:*100%*;width:*auto*;" class=*"card-img-top rounded mx-auto d-block m-2"* alt=*"img"*>

<div class=*"card-body"* >

<a href=*"product.jsp?productId=*<%=p.getProductId() %>*"* style="text-decoration: *none*;color:*black*;"> <h5 class=*"card-title"* ><%=p.getProductTitle() %></h5></a>

<p class=*"card- text"*><%=Helper.get10Words(p.getProductDescription()) %></p>

</div>

<div class=*"card-footer text-center"*>

<p style="font-size:*25px*"><span class=*"ml- 2"*><b>&#8377;<%=Helper.getProductSellingPrice(p.getProductPrice(), p.getProductDiscount()) %>/-</b></span>

<span class=*"ml-2"* style="font- size:*20px*;color:*red*"><s>&#8377;<%=p.getProductPrice()%></s></span>

<span class=*"ml-2"* style="font- size:*20px*;color:*green*"><%=p.getProductDiscount() %>&#37 off</span>

</p>

<span class=*"ml-2"* style="font-

size:*20px*;">Stock :</span>

<span class=*"ml-1"* style="font-

size:*20px*"><%**if**(p.getProductQuantity()<1){ %><span style="color:*red*;"><b><%=stock%></b></span><%} **else**{

%><%=p.getProductQuantity()%><% } %></span>

</div>

</div>

<%

}

%>

</div>

</div>

</div>

</div>

<%

}

%>

<!-- \*\*\*\*\*\*\* -->

</div>

</div>

</body>

</html>

## Implementation

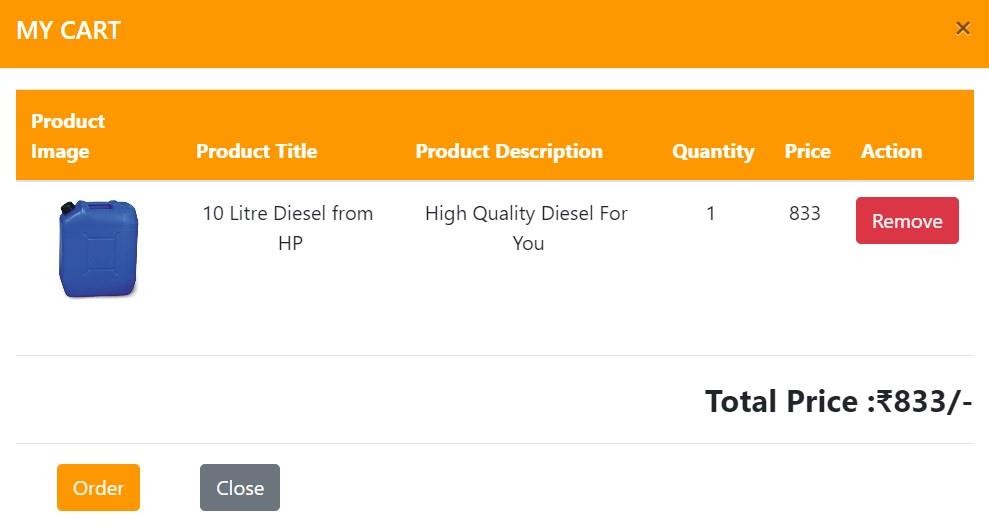


Fig 8.1 addtocart Feature

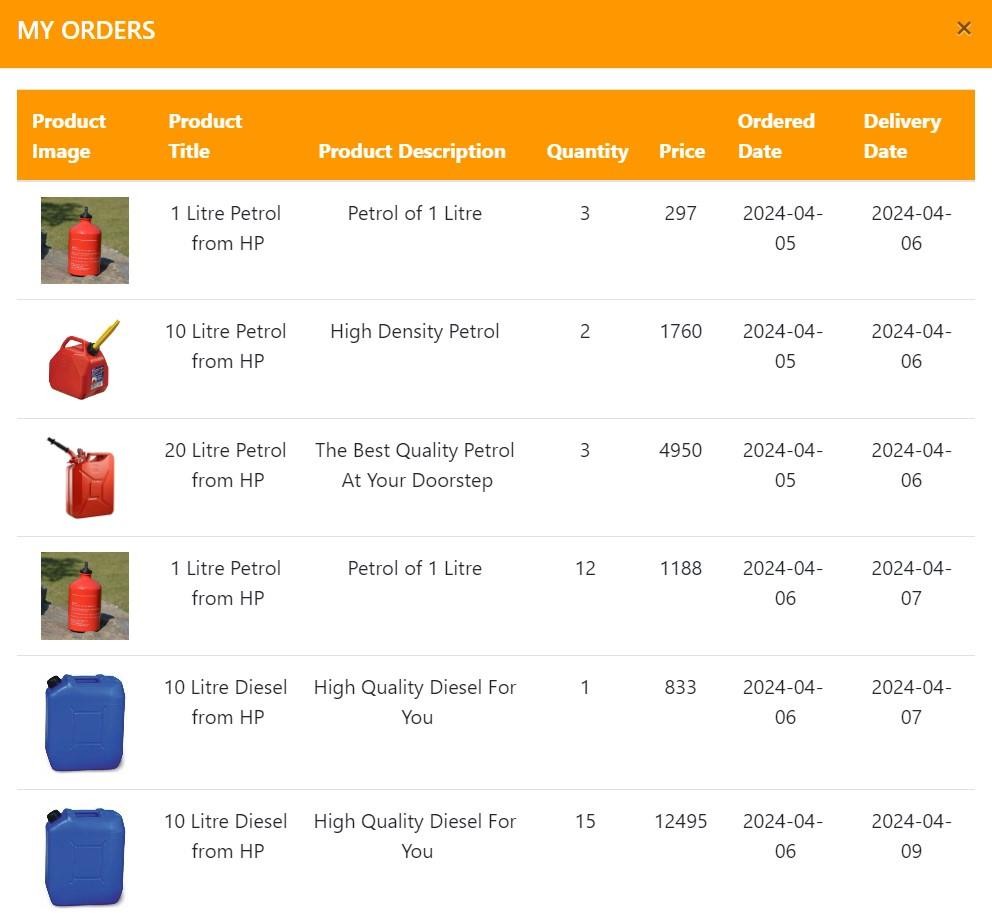


Fig 8.2 order history

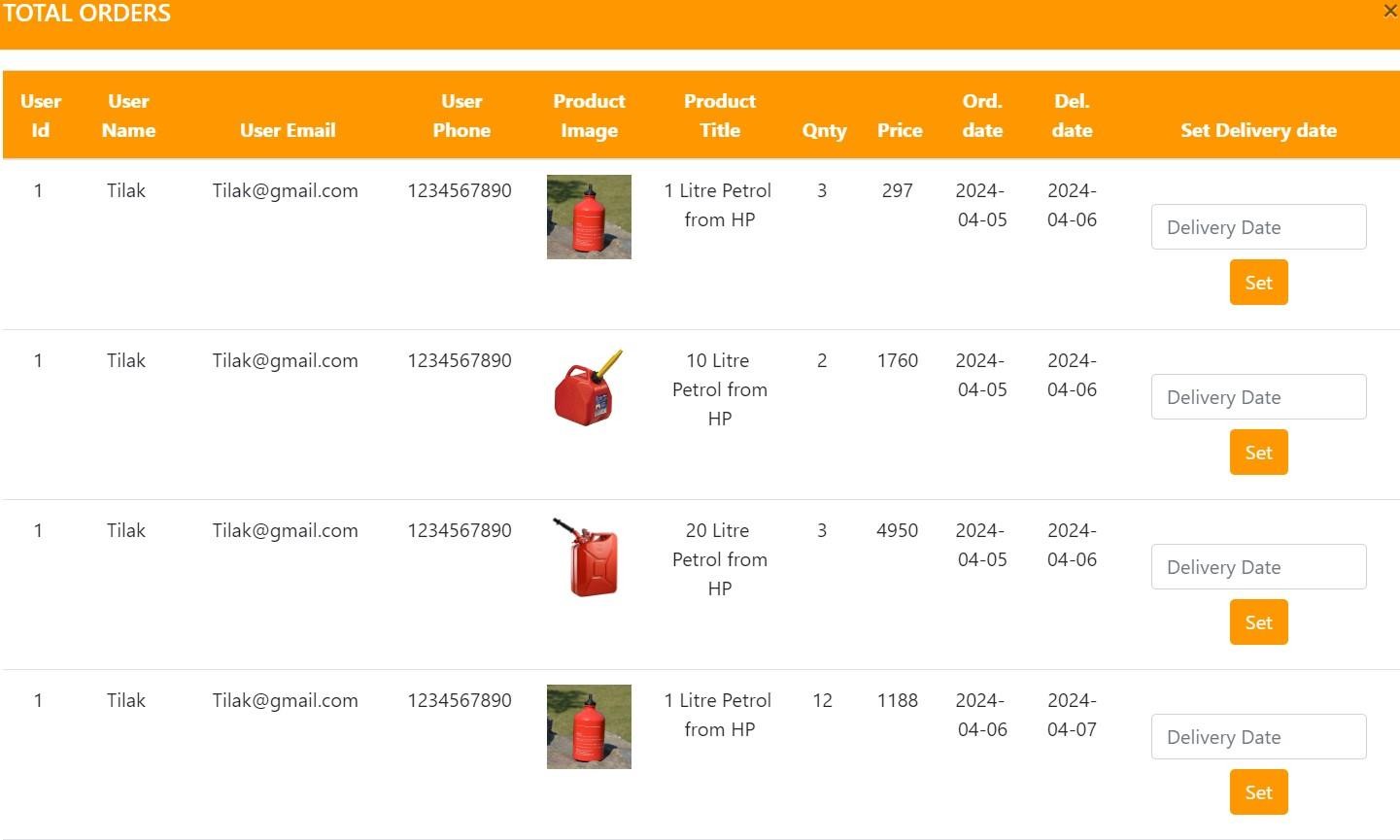


Fig 8.3 final output

* 1. **Conclusion**

# CHAPTER 9 CONCLUSION

The project presented showcases a comprehensive web application, eFuel, designed to revolutionize the fuel ordering process for homes. Through user-friendly interfaces and seamless integration with fuel suppliers, eFuel offers convenience, efficiency, and transparency to its users. Customers can now order fuel from the comfort of their homes, eliminating the need to visit physical stations and saving valuable time. With real-time pricing and availability information, users can make informed decisions, while the platform's customization options cater to individual preferences. Additionally, eFuel provides secure transactions, order tracking, and efficient management tools for administrators. Leveraging technologies such as Java Servlets, MySQL, and Apache Tomcat, this project demonstrates a scalable and robust solution for modernizing the fuel delivery industry. Overall, eFuel represents a significant advancement in fuel ordering systems, offering a user-centric approach that enhances convenience, accessibility, and user experience.

# BIBLIOGRAPHY

* + 1. R, Rajesh, Jesus Sandal Vinibha G, Kalaimathi K, Kamalakkanni P, and Kamatchi V. "NFC Identification System for Fuel Management." *SIJ Transactions on Computer Networks & Communication Engineering* 07, no. 04 (August 13, 2019): 01–06. [http://dx.doi.org/10.9756/sijcnce/v7i4/05020060102.](http://dx.doi.org/10.9756/sijcnce/v7i4/05020060102)
    2. Schala, Roland, and Maximilian Euringer. "Fuel System." *ATZextra worldwide* 12, no. 1 (September 2007): 156– 59. [http://dx.doi.org/10.1365/s40111-007-0030-1.](http://dx.doi.org/10.1365/s40111-007-0030-1)
    3. Schala, Roland, Michael Huber, and Harald Hagen. "Fuel System." *ATZextra worldwide* 13, no. 2 (June 2008): 94– 95. [http://dx.doi.org/10.1365/s40111-008-0064-z.](http://dx.doi.org/10.1365/s40111-008-0064-z)
    4. Revink, Ingo, Roger Letzer, and Thomas Just. "Fuel System." *ATZextra worldwide* 15, no. 11 (January 2010): 66– 69. [http://dx.doi.org/10.1365/s40111-010-0240-9.](http://dx.doi.org/10.1365/s40111-010-0240-9)
    5. MILEWSKI, Jaroslaw, and Krzysztof BADYDA. "E108 TRI-GENERATION SYSTEMS BASED ON HIGHTEMPERATURE FUEL CELLS(Distributed Energy System-2)." *Proceedings of the International Conference on Power Engineering (ICOPE)* 2009.1 (2009): \_1–275\_—\_1–279\_. [http://dx.doi.org/10.1299/jsmeicope.2009.1.\_1-275\_.](http://dx.doi.org/10.1299/jsmeicope.2009.1._1-275_)
    6. Chen, Yen-Jen, and Chia-Hung Chien. "Fuel Consumption System." *Journal of Computer and Communications* 03, no. 05 (2015): 153–58. [http://dx.doi.org/10.4236/jcc.2015.35019.](http://dx.doi.org/10.4236/jcc.2015.35019)
    7. Theobald, Jörg, Kay Schintzel, Andreas Krause, and Ulrich Doerges. "Fuel Injection System." *MTZ worldwide* 72, no. 4 (March 11, 2011): 4–9. [http://dx.doi.org/10.1365/s38313-011-0034-0.](http://dx.doi.org/10.1365/s38313-011-0034-0)
    8. Pala, Rameez Hassan. "Fuel Cell System and Their Technologies: A Review." *International Journal of Trend in Scientific Research and Development* Volume-3, Issue-2 (February 28, 2019): 153–58. [http://dx.doi.org/10.31142/ijtsrd20316.](http://dx.doi.org/10.31142/ijtsrd20316)
    9. Brun, C. Le, E. Godoy, D. Beauvois, G. Le Pache, and R. Noguera. "Modeling and Analysis of a Turbojet Fuel System." *International Journal of Computer Theory and Engineering* 6, no. 3 (2014): 260–66. [http://dx.doi.org/10.7763/ijcte.2014.v6.872.](http://dx.doi.org/10.7763/ijcte.2014.v6.872)
    10. Staiger, Robert, and Adrian Tantau. "Fuel Cell Heating System a Meaningful Alternative to Today’s Heating Systems." *Journal of Clean Energy Technologies* 5, no. 1 (2017): 35–41. <http://dx.doi.org/10.18178/jocet.2017.5.1.340>