SHOP HOUSE

A Report Submitted
In Partial Fulfillment of the Requirements
for the Degree of

Bachelor of Technology

In
COMPUTER SCIENCE & ENGINEERING
By

MOHD AMAAN KHAN
(2000560100068)

FAIZ MOHAMMAD
(2000560100045)

SAHIL KHAN
(2000560100086)

Under the Supervision of
Mr. WASEEM AHMAD
ASSISTANT PROFFESSOR

BBDNIIT

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
BABU BANARASI DAS NORTHERN INDIA INSTITUTE OF TECHNOLOGY,
LUCKNOW
Affiliated
To the

DR.APJ ABDUL KALAM TECHNICAL UNIVERSITY
LUCKNOW

(Formerly Uttar Pradesh Technical University)
Lucknow

May, 2024

Vision of the Institute

To establish a multi-disciplinary environment with excellence in technical education and research for developing competent professionals who meet the challenges of industrial and societal development with human values and ethics.

Mission of the Institute

M1. To provide an excellent environment with supporting infrastructure to prepare globally competent professionals acceptable to industry and society.

M2. To inculcate a spirit of research, innovation and entrepreneurship by exposing multi-disciplinary approach.

M3. To motivate aspiring graduates to solve real life problems with zeal of lifelong learning.

M4. To imbibe a healthy environment which helps to develop intellectual capabilities among graduates to transform them into professionals with human values and ethics.

BABU BANARASI DAS
NORTHERN INDIA INSTITUTE OF TECHNOLOGY

Affiliated to Dr. A.P.J. Abdul Kalam Technical University (AKTU College Code: 056)
Approved by All India Council for Technical Education (AICTE)
Sector II, Dr Akhilesh Das Nagar, Ayodhaya Road, Lucknow (UP) – India, 226028
Website: www.bbdniit.ac.in / headcse@bbdniit.ac.inPhone No.: +91-(522) - 6196222 / 6196223 / 6196336

Department of Computer Science and Engineering

Vision of the Department

To provide a conducive environment for learning and create research opportunities in the field of Computer Science and Engineering to meet global requirements using the latest technologies with ethical values.

Mission of the Department

M1. To provide theoretical and practical concepts across the Computer Science discipline.

M2. To impart an education that is well suited to meet the challenging software needs of the industry

M3. To nurture the undergraduate students with multi-disciplinary and research activities to grow as professionals.

M4. To empower students with moral and ethical values.

Program Educational Objective (PEOs)

The Program Educational Objectives of the U.G. program in the Department of Computer Science and Engineering are:

After 4 years of completing this program, the students:

PEO1: To encourage students to strengthen their technical capabilities for providing software solutions.

PEO2: To groom graduates as professional engineers to work with leadership and problem solving skills.

PEO3: To motivate students to work in multidisciplinary environment to fulfill industrial and societal needs.

Program Specific Objectives (PSOs)

PSO1: Ability to design and develop solutions for societal problems by using emerging technologies and standardized emerging principles.

PSO2: Develop and understanding for conceptual and practical aspects of programming languages with databases and develop solutions using suitable data structures and algorithmic techniques.

Program Outcomes (POs)

PO1 Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

PO2 Problem Analysis: Identify, formulate, review literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural science and engineering sciences.

PO3 Design / Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety and the cultural, societal and environmental considerations.

PO4 Conduct Investigations of Complex problems: Use research based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5 Modern Tool Usage: Create, Select, and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6 The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7 Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental context and demonstrate the knowledge of, and need for sustainable development.

PO8 Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9 Individual and Team Work: Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary settings.

PO10 Communication: Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11 Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12 Life Long Learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadcast of technological change.

Course Outcome

CO1. Analyze and understand the real-life problem and apply their knowledge to getprogramming solution **[K4,K5]**

CO2. Engage in the creative design process through the integration and application of diverse technical knowledge and expertise to meet customer needs and address social issues. **[K4,K5]**

CO3. Use the various tools and techniques, coding practices for developing real life solution to the problem.**[K5,K6]**

CO4. Find out the errors in software solutions and establishing the process to design maintainable software applications.**[K4, K5]**

CO5. Write the report about what they are doing in project and learning the team working skills. **[K5,K6]**

BABU BANARASI DAS
NORTHERN INDIA INSTITUTE OF TECHNOLOGY

Affiliated to Dr. A.P.J. Abdul Kalam Technical University (AKTU College Code: 056)
Approved by All India Council for Technical Education (AICTE)
Sector II, Dr Akhilesh Das Nagar, Ayodhaya Road, Lucknow (UP) – India, 226028
Website: www.bbdniit.ac.in / headcse@bbdniit.ac.inPhone No.: +91-(522) - 6196222 / 6196223

DECLARATION

I here by declare that the work presented in this report entitled "**SHOP HOUSE**", was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Date :

MOHD AMAAN KHAN
(2000560100068)

FAIZ MOHAMMAD
(2000560100045)

SAHIL KHAN
(2000560100086)

BBDNIIT

BABU BANARASI DAS
NORTHERN INDIA INSTITUTE OF TECHNOLOGY

Affiliated to Dr. A.P.J. Abdul Kalam Technical University (AKTU College Code: 056)
Approved by All India Council for Technical Education (AICTE)
Sector II, Dr Akhilesh Das Nagar, Ayodhaya Road, Lucknow (UP) – India, 226028
Website: www.bbdniit.ac.in / headcse@bbdniit.ac.inPhone No.: +91-(522) - 6196222 / 6196223

Department of Computer Science and Engineering

Department Vision	***Department Mission***
To provide a conducive environment for learning and create research opportunities in the field of Computer Science and Engineering to meet global requirements using the latest technologies with ethical values.	M1. To provide theoretical and practical concepts across the Computer Science discipline. M2. To impart an education that is well suited to meet the challenging software needs of the industry M3. To nurture the undergraduate students with multi-disciplinary and research activities to grow as professionals. M4. To empower students with moral and ethical values.

CERTIFICATE FROM DEPARTMENT

This is to certify that the project report entitled **SHOP HOUSE** submitted by

1. MR. MOHD AMAAN KHAN (2000560100068)

2. MR. FAIZ MOHAMMAD (2000560100045)

3. MR. SAHIL KHAN (2000560100086)

to the Department of Computer Science and Engineering, Babu Banarasi Das Northern India Institute of Technology, Lucknow in partial fulfillment of the requirements for the award of the Degree Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING is a bonafide record of work carried out by him/her under my/our guidance and supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute for the award of any Degree.

Date:

Mr. WASEEM AHMAD
(Assistant Professor-CSE)

Dr. ANURAG SHRIVASTAVA
(Professor & Head-CSE)

BABU BANARASI DAS

NORTHERN INDIA INSTITUTE OF TECHNOLOGY

Affiliated to Dr. A.P.J. Abdul Kalam Technical University (AKTU College Code: 056)
Approved by All India Council for Technical Education (AICTE)
Sector II, Dr Akhilesh Das Nagar, Ayodhaya Road, Lucknow (UP) – India, 226028
Website: www.bbdniit.ac.in / headcse@bbdniit.ac.inPhone No.: +91-(522) - 6196222 / 6196223

Department of Computer Science and Engineering

Department Vision	*Department Mission*
To provide a conducive environment for learning and create research opportunities in the field of Computer Science and Engineering to meet global requirements using the latest technologies with ethical values.	M1. To provide theoretical and practical concepts across the Computer Science discipline. M2. To impart an education that is well suited to meet the challenging software needs of the industry M3. To nurture the undergraduate students with multi-disciplinary and research activities to grow as professionals. M4. To empower students with moral and ethical values.

EVALUATION CERTIFICATE

This project report entitled **SHOP HOUSE** submitted by

1. MR. MOHD AMAAN KHAN (2000560100068)

2. MR. FAIZ MOHAMMAD (2000560100045)

3. MR. SAHIL KHAN (2000560100086)

is evaluated for the award of the Degree Bachelor of Technology in **Computer Science and Engineering.**

Date:

Place: LUCKNOW

Mr. WASEEM AHMAD
(Assistant Professor -CSE)

Dr. ANURAG SRIVASTAVA
(Professor & Head -CSE)

External Examiner
(Association)

TABLEOFCONTENTS Page no.

List of Figures

INTEODUCTION

Welcome to **Shop House** your premier destination for high-quality automobile parts and accessories! Whether you're a car enthusiast looking to upgrade your ride or a professional mechanic in need of reliable replacement parts, we've got everything you need to keep your vehicle running smoothly.

Shop House we understand the importance of having access to top-notch parts that you can rely on. That's why we've curated a comprehensive selection of products from trusted brands known for their exceptional quality and performance. From engine components to brake pads, from suspension systems to interior accessories, we've got it all right here at your fingertips.

But we're more than just a parts store. We're a team of automotive enthusiasts dedicated to helping you find the perfect solutions for your vehicle. Whether you're tackling a DIY project or need expert advice on which parts to choose, our knowledgeable staff is here to assist you every step of the way.

1.1 Objective

The objective of an automobile parts e-commerce website can be multifaceted, but generally, it revolves around providing a convenient, reliable, and comprehensive platform for customers to purchase automotive parts and accessories.

Convenience: Provide customers with a convenient online platform where they can easily browse, search for, and purchase a wide range of automobile parts and accessories from the comfort of their own homes or workshops.

Accessibility: Ensure accessibility to a diverse selection of high-quality automotive parts and accessories, catering to various makes, models, and years of vehicles, covering everything from routine maintenance components to specialized performance upgrades.

Reliability: Offer customers access to genuine, OEM (Original Equipment Manufacturer) parts as well as reputable aftermarket parts from trusted brands, ensuring reliability, durability, and compatibility with their vehicles.

Information and Guidance: Provide comprehensive product information, including specifications, compatibility details, installation guides, and customer reviews, to help customers make informed purchasing decisions.

Competitive Pricing and Deals: Provide competitive pricing on automotive parts and accessories, along with special deals, discounts, and promotions to attract and retain customers.

2.1 **WEB – DEVELOPMENT**

Web development is a broad term for the work involved in developing a web site for the Internet (World Wide Web) or an intranet (a private network). Web development can range from developing the simplest static single page of plain text to the most complex web-based internet applications, electronic businesses, and social network services. A more comprehensive list of tasks to which web development commonly refers, may include web engineering, web design, web content development, client liaison, client-side/side scripting. web server and network security configuration, and e-commerce development. Among web professionals, "web development" usually refers to the main non-design aspects of building web sites: writing markup and coding. Most recently Web development has come to mean the creation of content management systems or CMS. These CMS can be made from scratch, proprietary or open source. In broad terms the CMS acts as middleware between the database and the user through the browser. A principle benefit of a CMS is that it allows non-technical people to make changes to their web site without having technical knowledge.

For larger organizations and businesses, web development teams can consist of hundreds of people (web developers) and follow standard methods like Agile methodologies while developing websites. Smaller organizations may only require a single permanent or contracting developer, or secondary assignment to related job positions such as a graphic designer or information systems technician. Web development may be a collaborative effort between departments rather than the domain of a designated department. There are three kind of web developer specialization: front-end developer, back-end developer, and full-stack developer.

2.2 Web Site

A website is a collection of related web pages, including multimedia content, typically identified with a common domain name, and published on at least one web server. A website may be accessible via a public Internet Protocol (IP) network, such as the Internet, or a private local area network (LAN), by referencing a uniform resource locator (URL) that identifies the site.

Websites have many functions and can be used in various fashions; a website can be a personal website, a commercial website for a company, a government website or a non-profit organization website. Websites are typically dedicated to a particular topic or purpose, ranging from entertainment and social networking to providing news and education.

All publicly accessible websites collectively constitute the World Wide Web, while private websites, such as a company's website for its employees, and are typically a part of an intranet.

Web pages, which are the building blocks of websites, are documents, typically composed in plain text interspersed with formatting instructions of Hypertext Markup Language (HTML, XHTML). They may incorporate elements from other websites with suitable markup anchors. Web pages are accessed and transported with the Hypertext Transfer Protocol (HTTP), which may optionally employ encryption (HTTP Secure, HTTPS) to provide security and privacy for the user. The user's application, often a web browser, renders the page content according to its HTML markup instructions onto a display terminal. Hyperlinking between web pages conveys to the reader the site structure and guides the navigation of the site, which often starts with a home page containing a directory of the site web content. Some websites require user registration or subscription to access content. Examples of subscription websites include many business sites, news websites, academic journal websites, gaming websites, file-sharing websites, message boards, web-based email, social networking websites, websites providing real-time stock market data, as well as sites providing various other services. As of 2016 end users can access websites on a range of devices, including desktop and laptop computers, tablet computers, smartphones and smart TVs.

A web site consists of web pages which are interconnected to each other and contain various data and functionalities.

2.2 Web-Page

A web page, or webpage, is a document that is suitable for the World Wide Web and web browsers. A web browser displays a web page on a monitor or mobile device. The web page is what displays, but the term also refers to a computer file, usually written in HTML or comparable markup language. Web browsers coordinate the various web resource elements for the written web page, such as style sheets, scripts, and images, to present the web page. Typical web pages provide hypertext that includes a navigation bar or a sidebar menu to other web pages via hyperlinks, often referred to as links. On a network, a web browser can retrieve a web page from a remote web server. On a higher level, the web server may restrict access to only a private network such as a corporate intranet or it provides access to the World Wide Web. On a lower level, the web browser uses the Hypertext Transfer Protocol (HTTP) to make such requests. A static web page is delivered exactly as stored, as web content in the web server's file system, while a dynamic web page is generated by a web application that is driven by server- side software or client-side scripting. Dynamic website pages help the browser (the client) to enhance the web page through user input to the server.

THE STEPS TO CREATE A WEB SITE

- Creating a web site requires multiple steps which includes the following:
- Creating a UI(User interface) Scripting(Both at server end and client end)
- Creating a backend or the database

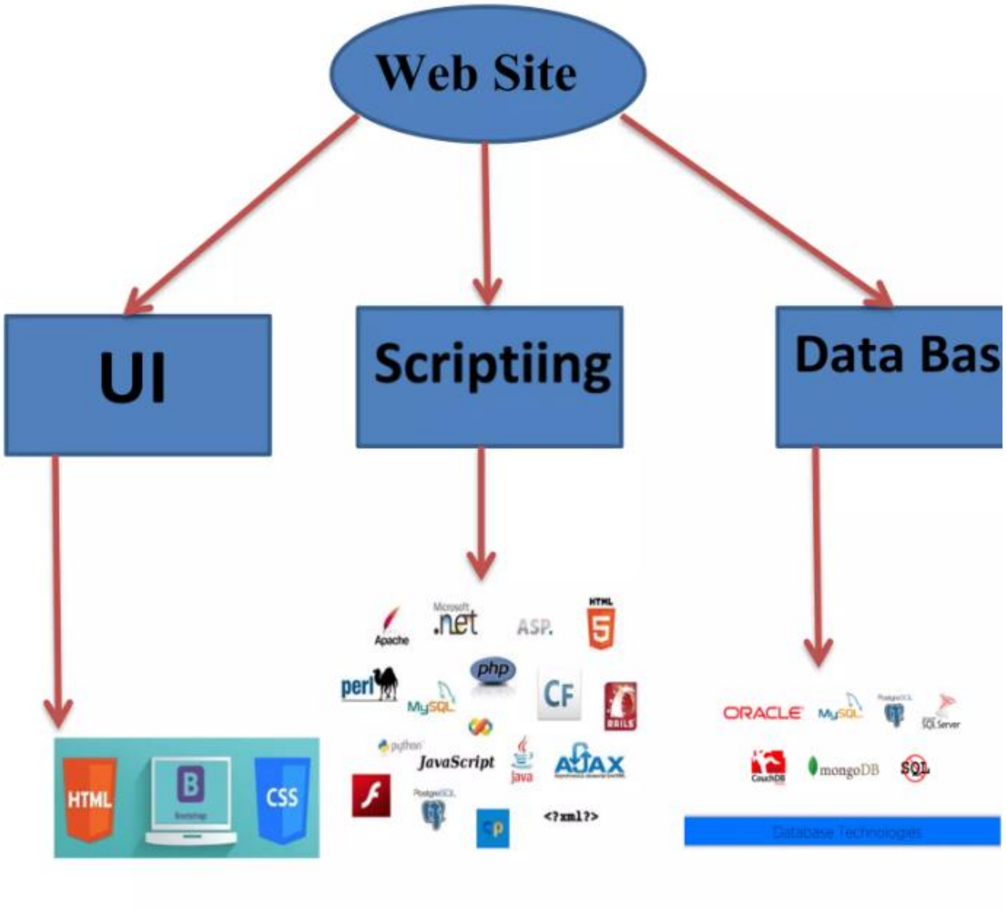


Fig 3.1

3.1 UI DEVELOPMENT

Technologies that are mostly used to develop a User Interface are:

- HTML
- CSS
- Bootstrap.

3.1.1 HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a webserver or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> introduce content into the page directly. Others such as <p>...</p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

HTML markup consists of several key components, including those called tags (and their attributes), character-based data types, character references and entity references. HTML tags most commonly come in pairs like <h1> and </h1>, although some represent empty elements and so are unpaired, for example <img>. The first tag in such a pair is the start tag, and the second is the end tag.

Another important component is the HTML document type declaration, which triggers standards mode rendering.

The following is an example of the classic Hello world program, a common test employed for comparing programming languages, scripting languages and markup languages. This example is made using 9 lines of code:

General Syntax of HTML

```
<!DOCTYPE html>
<html>
    <head>
          <title>This is a title</title>
   </head>
<body>
      <p>Hello world!</p>
</body>
</html>
```

(The text between <html> and </html> describes the web page, and the text between <body> and </body> is the visible page content. The markup text "<title>This is a title</title>" defines the browser page title.)

3.1.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices.

It can also display the web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified.The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

Types of CSS:

- Inline CSS:

In this CSS is applied in between the tags

Eg: <tag style="styling">Hello World</tag>

- Internal CSS:

In this Theess code is defined inside the style tag in the head section of the HTML page.

General Syntax:

```
<html>
 <head>
     <style>
       <!-- CSS STYLING ->
     </style>
 </head>
</html>
```

- External CSS:

In this the CSS code is written on another page and is linked to the HTML page. It is advantageous to use this type of styling as we can use the same file to style various HTML pages.

External CSS uses the extension css and is applied using the following syntax:

```
            <html>
                    <head>
                            <link relation="stylesheet" type="css" href="url to the page">
                    </head>
          </html>
```

All the CSS style types are important but can be used in different situations.

- Inline CSS is used when only small changes are to be done to the HTML tag and the changes are to be reflected only to that specific tag

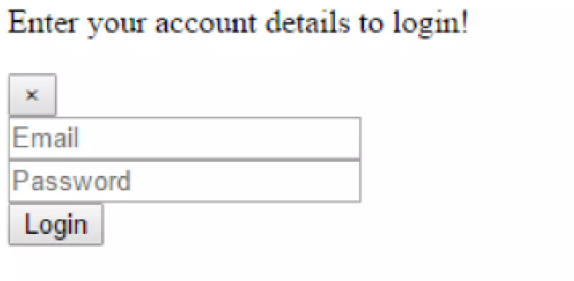
- Internal CSS is used when the individual HTML pages have to be designed differently. This also slows the page load system if the internal styling is long.

- External CSS files are maintained to design multiple pages and use common styles over various pages. It is useful as it helps in managing the resources in an easy manner.

Both HTML and CSS are used to create a UI but CSS behaves like a makeup on the face of an actress which makes her look even more beautiful than she is in reality.

And here is the difference:

Before using CSS in HTML page:



Enter your account details to login!

Fig 3.2

After using CSS in HTML Page:

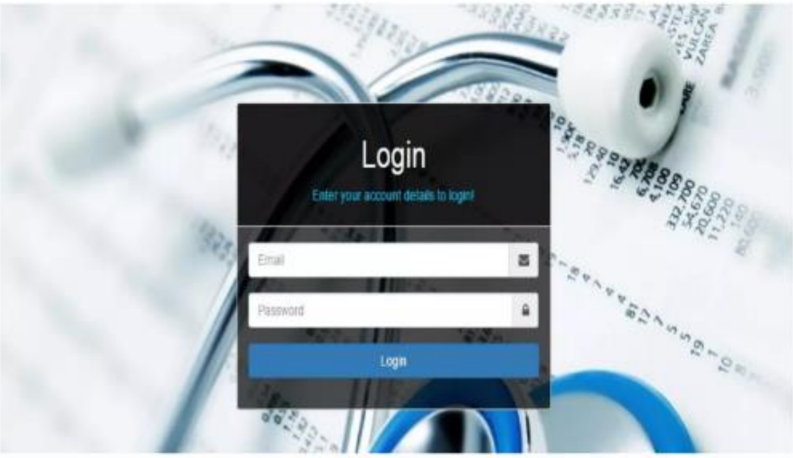


Fig 3.3

3.1.3 BOOTSTRAP

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML and CSS-based design templates for typography. forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap is the second most-starred project on GitHub, with more than 107,000 stars and 48,000 forks.

Bootstrap, originally named Twitter Blueprint, was developed by Mark Otto and Jacob Thornton at Twitter as a framework to encourage consistency across internal tools. Before Bootstrap, various libraries were used for interface development, which led to inconsistencies and a high maintenance burden. According to twitter developer Mark Otto:

"A super small group of developers and I got together to design and build a new internal tool and saw an opportunity to do something more. Through that process, we saw ourselves build something much more substantial than another internal tool. Months later, we ended up with an early version of Bootstrap as a way to document and share common design patterns and assets within the company."

After a few months of development by a small group, many developers at Twitter began to contribute to the project as a part of Hack Week, a hackathon-style week for the Twitter development team. It was renamed from Twitter Blueprint to Bootstrap, and released as an open source project on August 19, 2011. It has continued to be maintained by Mark Otto, Jacob Thornton, and a small group of core developers, as well as a large community of contributors.

On January 31, 2012, Bootstrap 2 was released, which added a twelve-column responsive grid layout system, inbuilt support for Glyphicons, several new components, as well as changes to many of the existing components.On August 19, 2013, Bootstrap 3 was released, which redesigned components to use flat design, and a mobile first approach. Bootstrap 3 supports the latest versions of the Google Chrome, Firefox, Internet Explorer, Opera, and Safari (except on Windows). It additionally supports back to IE8 and the latest Firefox Extended Support Release (ESR). Since 2.0, Bootstrap supports responsive web design. This means the layout of web pages adjusts dynamically, taking into account the characteristics of the device used (desktop, tablet, mobile phone).

Starting with version 3.0, Bootstrap adopted a mobile-first design philosophy, emphasizing responsive design by default.

The version 4.0 alpha release added Sass and flexbox support

Installing and linking bootstrap to the HTML page:

Install bootstrap from https://getbootstrap.com/

Copy the bootstrap.min.css file to your CSS folder and link it to the HTML page in the similar manner to how any other CSS file is linked. Link the bootstrap.min.js file which is present in the JS folder of the bootstrap. It canbe linked using script tag.

Eg: <script src="url to bootstrap.min.js"></script>

Now use bootstrap classes to reduce the work of designing which was earlier done through CSS.

3.2 SCRIPTING

There are two scripting methodologies.

1. Server side scripting: This scripting is done at the server end

2. Client side scripting: This scripting is done at the client end or the browser.

3.2.1 SERVER SIDE SCRIPTING

Server-side scripting is a technique used in web development which involves employing scripts on a web server which produce a response customized for each user's (client's) request to the website. The alternative is for the web server itself to deliver a static web page. Scripts can be written in any of a number of server-side scripting languages that are available (see below). Server-side scripting is distinguished from client-side scripting where embedded scripts, such as JavaScript, are run client-side in a web browser, but both techniques are often used together.

Server-side scripting is often used to provide a customized interface for the user. These scripts may assemble client characteristics for use in customizing the response based on those characteristics, the user's requirements, access rights, etc. Server-side scripting also enables the website owner to hide the source code that generates the interface, whereas with client-side scripting, the user has access to all the code received by the client. A down-side to the use of server-side scripting is that the client needs to make further requests over the network to the server in order to show new information to the user via the web browser. These requests can slow down the experience for the user, place more load on the server, and prevent use of the application when the user is disconnected from the server.

When the server serves data in a commonly used manner, for example according to the HTTP or FTP protocols, users may have their choice of a number of client programs (most modern web browsers can request and receive data using both of those protocols). In the case of more specialized applications, programmers may write their own server, client, and communications protocol that can only be used with one another.

3.2.2 Server Side scripting Languages

There are several languages that can be used for server-side programming:

- PHP
- ASP.NET (C# OR Visual Basic)
- C++
- Java and JSP
- Python

- Ruby on Rails and so on.

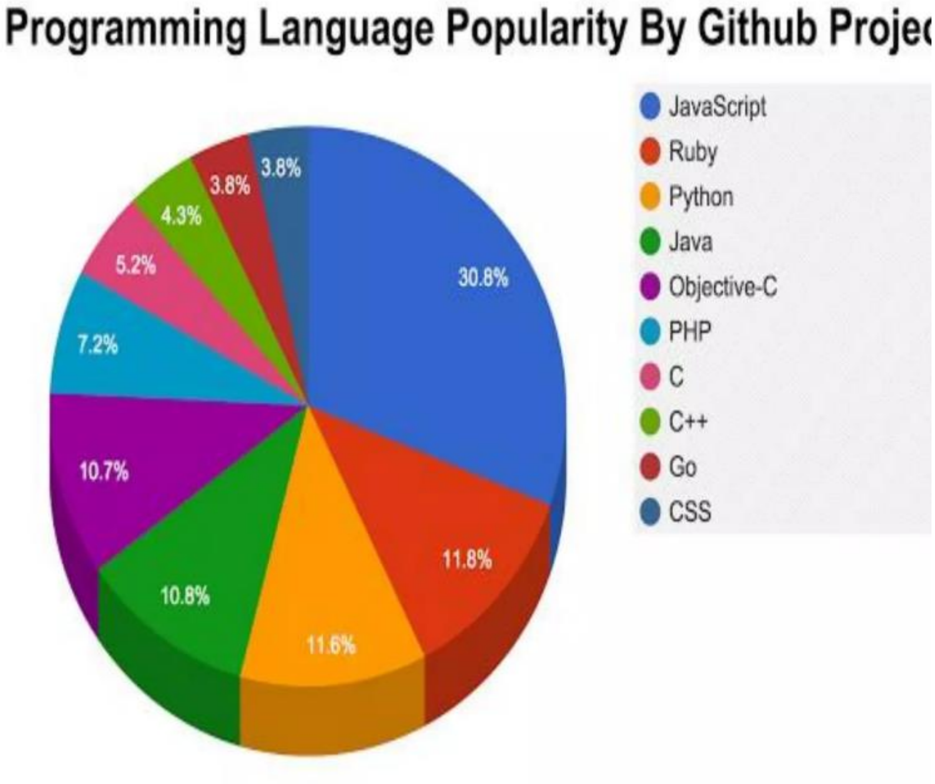


Fig 3.2.1

3.2.3 CLIENT SIDE SCRIPTING

Client-side scripting is changing interface behaviors within a specific web page in response to mouse or keyboard actions, or at specified timing events. In this case, the dynamic behavior occurs within the presentation. The client-side content is generated on the user's local computer system.

Such web pages use presentation technology called rich interfaced pages. Client-side scripting languages like JavaScript or ActionScript, used for Dynamic

HTML (DHTML) and Flash technologies respectively, are frequently used to orchestrate media types (sound, animations, changing text, etc.) of the presentation. Client-side scripting also allows the use of remote scripting, a technique by which the DHTML page requests additional information from a server, using a hidden frame, XML Http Requests, or a Web service.The first widespread use of JavaScript was in 1997, when the language was standardized as ECMAScript and implemented in Netscape 3.

Example:

The client-side content is generated on the client's computer. The web browser retrieves a page from the server, then processes the code embedded in the page (typically written in JavaScript) and displays the retrieved page's content to the user.

The most popularly used client side scripting languages is Java Script. Flow of request from browser to server:

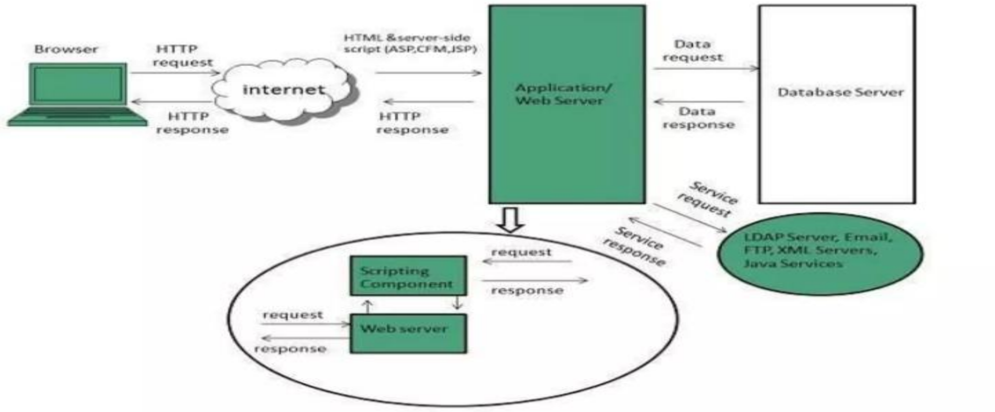


Fig 3.2.2

3.3 DATABASE

A **database** is an organized collection of data. It is the collection of schemas, tables, queries, reports, views, and other objects. The data are typically organized to model aspects of reality in a way that supports processes requiring information, such as modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

A **database management system (DBMS)** is a computer software application that interacts with the user, other applications, and the database itself to capture and analyze data. A general-purpose DBMS is designed to allow the definition, creation, querying, update, and administration of databases. Well-known DBMSs include MySQL, PostgreSQL, MongoDB,

MariaDB, Microsoft SQL Server, Oracle, Sybase, SAP HANA, MemSQL and IBM DB2. A database is not generally portable across different DBMSs, but different DBMS can interoperate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one DBMS. Database management systems are often classified according to the database model that they support; the most popular database systems since the 1980s have all supported the relational model as represented by the SQL language. Sometimes a DBMS is loosely referred to as a “database”.

CHAPTER-4

SOFTWARE REQUIREMENT SPECIFICATION

4.1 Hardware Requirements

The selection of hardware is very important in the existence and proper working of any software. When selecting hardware, the size and requirements are also important.

Processor	Intel core i5
RAM	**8.0 GB**
Hard Disk Drive	**500GB**

4.2 Software Requirements

Number	Description
1	Windows 8, 10 ,11
2	HTML/CSS/JavaScript/Bootstrap
3	Visual studio code

CHAPTER-5

DATA FLOW DIAGRAM

Data Flow Diagrams show the flow of data from external entities into the system, and from one process to another within the system. There are four symbols for drawing a DFD:

I. Rectangles representing external entities, which are sources or destinations of data .

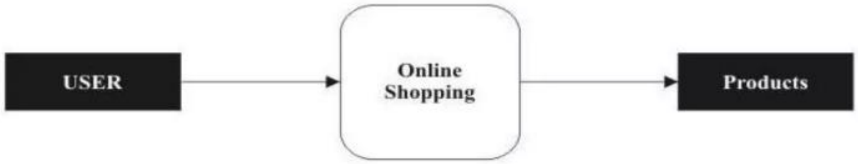
II. Ellipses representing processes, which take data as input, validate and process it and output it.

III. Arrows representing the data flows, which can either, be electronic data or physical items.

IV. Open-ended rectangles or a Disk symbol representing data stores, including electronic stores such as databases or XML files and physical stores such as filing cabinets or stacks of paper.

Figures below are the Data Flow Diagrams for the current system. Each process within the system is first shown as a Context Level DFD and later as a Detailed DFD. The Context Level DFD provides a conceptual view of the process and its surrounding input, output and data stores. The Detailed DFD provides a more detailed and comprehensive view of the interaction among the sub-processes within the system.

CONTEXT LEVEL DIAGRAM



DFD-1

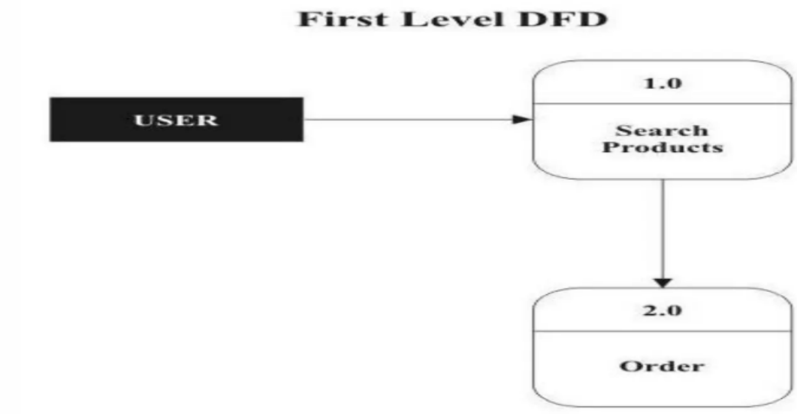


Fig5.1

DFD-2

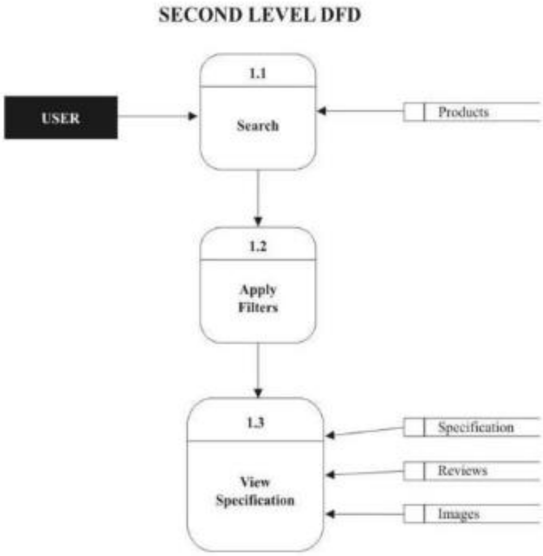


Fig 5.2

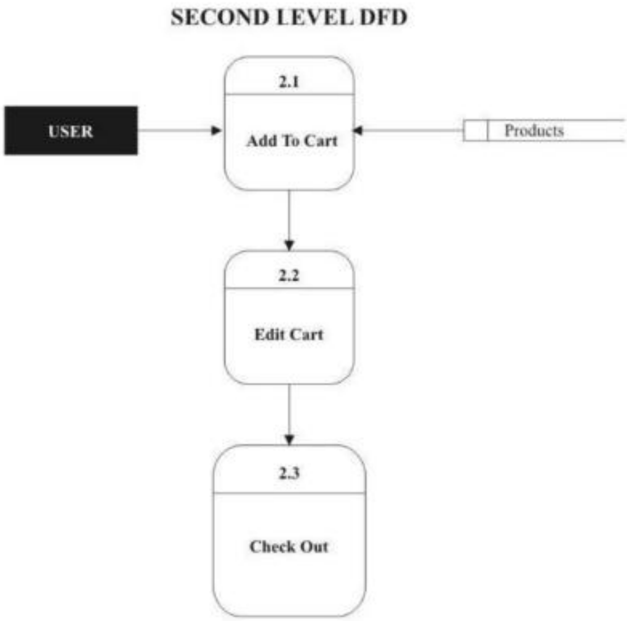


Fig 5.3

DFD-3

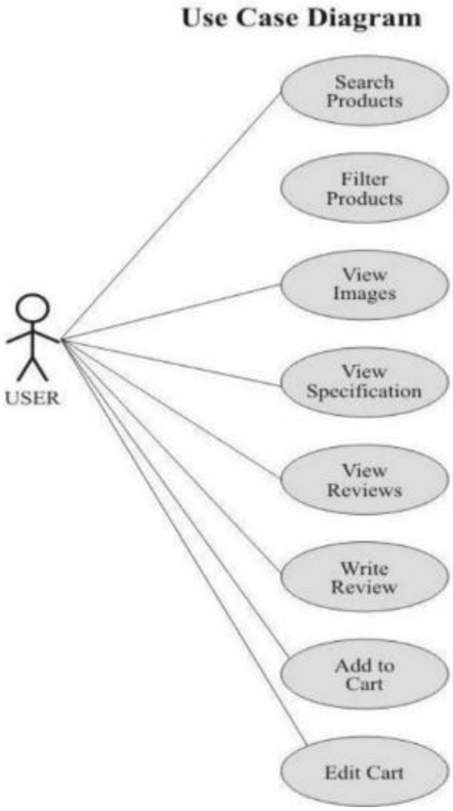


Fig 5.4

CHAPTER-6

PROJECT

6.1 PROJECT (Advanced Technologies):

Name: SHOP HOUSE

6.2 Technologies Used:

- HTML
- CSS
- Bootstrap
- Java Script
- Django

Team Size: 3

6.3 TECHNICAL DETAILS:

- Front end is designed using HTML, CSS and Bootstrap.
- Backend is based on Python using Django framework.
- The forms are made using the HTML, Bootstrap for designing.
- Admin panel are made using python and inbuilt library of Django.

6.4 CODE

BASE.html:

```
<!DOCTYPE html>
{% load static %}
<html lang="en">
  <head>
    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootst
rap.min.css" rel="stylesheet" integrity="sha384-
4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+Pt
moHDEXuppvnDJzQIu9" crossorigin="anonymous" />
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.10.5/font/bootstrap-icons.css" />
    {% comment %} <link rel="stylesheet" type="text/css"
href="{% static 'css/style.css' %}" />  {% endcomment %}

    <title>
      {% block xyz %}
      {% endblock %}
    </title>

       <style>
        body {
          background-image: url("{% static 'images/background.jpg'
%}");
          background-size: cover; /* Scale the image to cover the
entire screen */
          background-position: center; /* Center the image */
      }
       </style>
  </head>

  <body>
```

```
    <nav class="navbar navbar-expand-lg bg-warning">
      <div class="container-fluid">
        {% if request.session.name %}

          <a class="navbar-brand" href="#">{{ request.session.name }}</a>
        {% else %}

          <a class="navbar-brand" href="#">E-Shop</a>
        {% endif %}
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation"><span class="navbar-toggler-icon"></span></button>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
          <ul class="navbar-nav me-auto mb-2 mb-lg-0">
            <li class="nav-item">
              <a class="nav-link active" aria-current="page" href="{% url 'home' %}">Home</a>
            </li>
              </ul>
            </li>
          </ul>
          <form class="d-flex mx-2" role="search">
            <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search" />
            <button class="btn btn-outline-success" type="submit">Search</button>
          </form>
          <a href="{% url 'cart' %}" type="button" class="btn btn-primary position-relative ms-3">
            <i class="bi bi-cart"></i>
            <span class="position-absolute top-0 start-100 translate-middle badge rounded-pill bg-danger">
              {{ request.session.cart|length }}
              <span class="visually-hidden">unread messages</span>
            </span>
          </a>
          {% if request.session.name %}
            <a href="{% url 'logout' %}" type="button" class="btn
```

```
btn-success mx-2">Logout</a>
            <a href="{% url 'order' %}" type="button" class="btn btn-success">Order</a>
          {% else %}
            <button type="button" class="btn btn-success mx-2" data-bs-toggle="modal" data-bs-target="#login">Login</button>

            <button type="button" class="btn btn-success" data-bs-toggle="modal" data-bs-target="#signup">Sign up</button>
          {% endif %}
        </div>
      </div>

    </nav>

    {% block content %}
    {% endblock %}
    {% comment %}model for signup{% endcomment %}
    <!-- Modal -->
    <div class="modal fade" id="signup" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            <h1 class="modal-title fs-5" id="exampleModalLabel">Registration</h1>
            <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
          </div>
          <div class="modal-body">
            <form class="row g-3" action="{% url 'signup' %}" method="post">
              {% csrf_token %}
              <div class="col-md-6">
                <label for="fname" class="form-label">First Name</label>
                <input type="text" name="fname" class="form-control" id="fname" />
              </div>
              <div class="col-md-6">
                <label for="lname" class="form-label">Last Name</label>
                <input type="text" name="lname" class="form-control" id="lname" />
              </div>
              <div class="col-md-6">
                <label for="inputEmail4" class="form-label">Email</label>
                <input type="email" name="email" class="form-control" id="inputEmail4" />
              </div>
              <div class="col-md-6">
                <label for="inputPassword4" class="form-label">Password</label>
                <input type="password" name="pwd" class="form-control" id="inputPassword4" />

              </div>
               <div class="col-md-6">
                <label for="mobile" class="form-label">Mobile</label>
                <input type="number" name="mobile" class="form-control" id="mobile" />
              </div>
              <div class="col-md-6">
                <label for="inputState" class="form-label">Gender</label>
                <br />
                <input type="radio" name="gender" value="Male" />Male
                <br />
                <input type="radio" name="gender" value="Female" />Female
              </div>

              <div class="col-12">
                <button type="submit" class="btn btn-primary">Signup</button>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
    {% comment %}model for login{% endcomment %}
    <!-- Modal -->
    <div class="modal fade" id="login" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            <h1 class="modal-title fs-5" id="exampleModalLabel">Login</h1>
            <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
          </div>
          <div class="modal-body">
            <form action="{% url 'login' %}" method="post">
              {% csrf_token %}
              <div class="mb-3">


   <label for="formGroupExampleInput" class="form-label">Email</label>
                <input type="text" class="form-control" name="emailid" id="formGroupExampleInput" placeholder="enter your email..." />
              </div>
              <div class="mb-3">
                <label for="formGroupExampleInput2" class="form-label">Password</label>
                <input type="password" class="form-control" name="password" id="formGroupExampleInput2" placeholder="Enter your password" />
              </div>
  <div class="col-12">
                <button type="submit" class="btn btn-primary">Login</button>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
    <script>
```

```
      // JavaScript to display alert if login attempt fails
      {% if error_message %}
          alert("{{ error_message }}");
      {% endif %}
  </script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-HwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGAoxmnlMuBnhbgrkm" crossorigin="anonymous"></script>
  </body>
</html>
```

HOME.html

```
<!DOCTYPE html>
{% load static %}
<html lang="en">
  <head>
    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnDJzQIu9" crossorigin="anonymous" />
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.5/font/bootstrap-icons.css" />
    {% comment %} <link rel="stylesheet" type="text/css" href="{% static 'css/style.css' %}" />  {% endcomment %}

    <title>
      {% block xyz %}
      {% endblock %}
    </title>

       <style>
```

```
        body {
          background-image: url("{% static 'images/background.jpg'
%}");
          background-size: cover; /* Scale the image to cover the
entire screen */
          background-position: center; /* Center the image */
      }
       </style>
  </head>

  <body>
    <nav class="navbar navbar-expand-lg bg-warning">
      <div class="container-fluid">
        {% if request.session.name %}
          <a class="navbar-brand" href="#">{{ request.session.name
}}</a>

        {% else %}
          <a class="navbar-brand" href="#">E-Shop</a>
        {% endif %}
        <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation"><span class="navbar-toggler-
icon"></span></button>
        <div class="collapse navbar-collapse"
id="navbarSupportedContent">
          <ul class="navbar-nav me-auto mb-2 mb-lg-0">
            <li class="nav-item">
              <a class="nav-link active" aria-current="page" href="{%
url 'home' %}">Home</a>
            </li>
              </ul>
            </li>
          </ul>
          <form class="d-flex mx-2" role="search">
            <input class="form-control me-2" type="search"
placeholder="Search" aria-label="Search" />
            <button class="btn btn-outline-success"
type="submit">Search</button>
          </form>
          <a href="{% url 'cart' %}" type="button" class="btn btn-
```

```
primary position-relative ms-3">
            <i class="bi bi-cart"></i>
            <span class="position-absolute top-0 start-100 translate-middle badge rounded-pill bg-danger">
              {{ request.session.cart|length }}
              <span class="visually-hidden">unread messages</span>
            </span>
          </a>
          {% if request.session.name %}
            <a href="{% url 'logout' %}" type="button" class="btn btn-success mx-2">Logout</a>
            <a href="{% url 'order' %}" type="button" class="btn btn-success">Order</a>
          {% else %}
            <button type="button" class="btn btn-success mx-2" data-bs-toggle="modal" data-bs-target="#login">Login</button>
            <button type="button" class="btn btn-success" data-bs-toggle="modal" data-bs-target="#signup">Sign up</button>
          {% endif %}

        </div>
      </div>
    </nav>

    {% block content %}
    {% endblock %}
    {% comment %}model for signup{% endcomment %}
    <!-- Modal -->
    <div class="modal fade" id="signup" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            <h1 class="modal-title fs-5" id="exampleModalLabel">Registration</h1>
            <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
          </div>
          <div class="modal-body">
            <form class="row g-3" action="{% url 'signup' %}" method="post">
              {% csrf_token %}
              <div class="col-md-6">
                <label for="fname" class="form-label">First
Name</label>
                <input type="text" name="fname" class="form-control"
id="fname" />
              </div>
              <div class="col-md-6">
                <label for="lname" class="form-label">Last
Name</label>
                <input type="text" name="lname" class="form-control"
id="lname" />
              </div>
              <div class="col-md-6">
                <label for="inputEmail4" class="form-
label">Email</label>
                <input type="email" name="email" class="form-
control" id="inputEmail4" />
              </div>
              <div class="col-md-6">
                <label for="inputPassword4" class="form-
label">Password</label>

                <input type="password" name="pwd" class="form-
control" id="inputPassword4" />
              </div>

              <div class="col-md-6">
                <label for="mobile" class="form-label">Mobile</label>
                <input type="number" name="mobile" class="form-
control" id="mobile" />
              </div>
              <div class="col-md-6">
                <label for="inputState" class="form-
label">Gender</label>
                <br />
                <input type="radio" name="gender" value="Male"
/>Male
                <br />
                <input type="radio" name="gender" value="Female"
/>Female
              </div>
```

```
       <div class="col-12">
        <button type="submit" class="btn btn-primary">Signup</button>
       </div>
      </form>
     </div>
    </div>
   </div>
  </div>
  {% comment %}model for login{% endcomment %}
  <!-- Modal -->
  <div class="modal fade" id="login" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
   <div class="modal-dialog">
    <div class="modal-content">
     <div class="modal-header">
      <h1 class="modal-title fs-5" id="exampleModalLabel">Login</h1>
      <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
     </div>
     <div class="modal-body">
      <form action="{% url 'login' %}" method="post">

       {% csrf_token %}
       <div class="mb-3">
        <label for="formGroupExampleInput" class="form-label">Email</label>

        <input type="text" class="form-control" name="emailid" id="formGroupExampleInput" placeholder="enter your email..." />
       </div>
       <div class="mb-3">
        <label for="formGroupExampleInput2" class="form-label">Password</label>
        <input type="password" class="form-control" name="password" id="formGroupExampleInput2" placeholder="Enter your password" />
       </div>
       <div class="col-12">
        <button type="submit" class="btn btn-primary">Login</button>
       </div>
      </form>
     </div>
    </div>
   </div>
  </div>
  <script>
   // JavaScript to display alert if login attempt fails
   {% if error_message %}
     alert("{{ error_message }}");
   {% endif %}
 </script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-HwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGAoxmnlMuBnhbgrkm" crossorigin="anonymous"></script>
 </body>
</html>
```

Cart.html:

```
<!DOCTYPE html>
{% load static %}
<html lang="en">
 <head>
  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
```

```
4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+Pt
moHDEXuppvnDJzQIu9" crossorigin="anonymous" />
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.10.5/font/bootstrap-icons.css" />
    {% comment %} <link rel="stylesheet" type="text/css"
href="{% static 'css/style.css' %}" />  {% endcomment %}

    <title>
      {% block xyz %}
      {% endblock %}
    </title>

       <style>
        body {
          background-image: url("{% static 'images/background.jpg'
%}");
          background-size: cover; /* Scale the image to cover the
entire screen */
          background-position: center; /* Center the image */
      }
       </style>
  </head>

  <body>
    <nav class="navbar navbar-expand-lg bg-warning">
      <div class="container-fluid">
        {% if request.session.name %}
          <a class="navbar-brand" href="#">{{ request.session.name
}}</a>
        {% else %}
          <a class="navbar-brand" href="#">E-Shop</a>
        {% endif %}
        <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation"><span class="navbar-toggler-
icon"></span></button>
        <div class="collapse navbar-collapse"
id="navbarSupportedContent">
          <ul class="navbar-nav me-auto mb-2 mb-lg-0">
            <li class="nav-item">
```

```
              <a class="nav-link active" aria-current="page" href="{% url 'home' %}">Home</a>
            </li>
              </ul>
            </li>
          </ul>
          <form class="d-flex mx-2" role="search">
            <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search" />
            <button class="btn btn-outline-success" type="submit">Search</button>
          </form>
          <a href="{% url 'cart' %}" type="button" class="btn btn-primary position-relative ms-3">
            <i class="bi bi-cart"></i>
            <span class="position-absolute top-0 start-100 translate-middle badge rounded-pill bg-danger">
              {{ request.session.cart|length }}
              <span class="visually-hidden">unread messages</span>
            </span>
          </a>
          {% if request.session.name %}
            <a href="{% url 'logout' %}" type="button" class="btn btn-success mx-2">Logout</a>
            <a href="{% url 'order' %}" type="button" class="btn btn-success">Order</a>
          {% else %}
            <button type="button" class="btn btn-success mx-2" data-bs-toggle="modal" data-bs-target="#login">Login</button>
            <button type="button" class="btn btn-success" data-bs-toggle="modal" data-bs-target="#signup">Sign up</button>
          {% endif %}
        </div>


      </div>
    </nav>

    {% block content %}
    {% endblock %}
    {% comment %}model for signup{% endcomment %}
    <!-- Modal -->
    <div class="modal fade" id="signup" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            <h1 class="modal-title fs-5" id="exampleModalLabel">Registration</h1>
            <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
          </div>
          <div class="modal-body">
            <form class="row g-3" action="{% url 'signup' %}" method="post">
              {% csrf_token %}
              <div class="col-md-6">
                <label for="fname" class="form-label">First Name</label>
                <input type="text" name="fname" class="form-control" id="fname" />
              </div>
              <div class="col-md-6">
                <label for="lname" class="form-label">Last Name</label>
                <input type="text" name="lname" class="form-control" id="lname" />
              </div>
              <div class="col-md-6">
                <label for="inputEmail4" class="form-label">Email</label>
                <input type="email" name="email" class="form-control" id="inputEmail4" />
              </div>
              <div class="col-md-6">
                <label for="inputPassword4" class="form-label">Password</label>
                <input type="password" name="pwd" class="form-control" id="inputPassword4" />

              </div>

              <div class="col-md-6">
                <label for="mobile" class="form-label">Mobile</label>
                <input type="number" name="mobile" class="form-control" id="mobile" />
              </div>
              <div class="col-md-6">
                <label for="inputState" class="form-label">Gender</label>
                <br />
                <input type="radio" name="gender" value="Male" />Male
                <br />
                <input type="radio" name="gender" value="Female" />Female
              </div>

              <div class="col-12">
                <button type="submit" class="btn btn-primary">Signup</button>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
    {% comment %}model for login{% endcomment %}
    <!-- Modal -->
    <div class="modal fade" id="login" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            <h1 class="modal-title fs-5" id="exampleModalLabel">Login</h1>
            <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
          </div>
          <div class="modal-body">
            <form action="{% url 'login' %}" method="post">
              {% csrf_token %}
              <div class="mb-3">
                <label for="formGroupExampleInput" class="form-label">Email</label>
        <input type="text" class="form-control"
name="emailid" id="formGroupExampleInput"
placeholder="enter your email..." />
       </div>
       <div class="mb-3">
        <label for="formGroupExampleInput2" class="form-
label">Password</label>
        <input type="password" class="form-control"
name="password" id="formGroupExampleInput2"
placeholder="Enter your password" />
       </div>
     <div class="col-12">
        <button type="submit" class="btn btn-
primary">Login</button>
       </div>
      </form>
     </div>
    </div>
   </div>
  </div>
  <script>
   // JavaScript to display alert if login attempt fails
   {% if error_message %}
     alert("{{ error_message }}");
   {% endif %}
 </script>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstra
p.bundle.min.js" integrity="sha384-
HwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGA
oxmnlMuBnhbgrkm" crossorigin="anonymous"></script>
 </body>
</html>
```

Order.html:

```
{% extends "base.html" %}
{% load cart %}

{% block title %}

Order
{% endblock title %}



{% block content %}
<div class="container my-3">
   <table class="table table-success table-striped">
       <tr class="text-center">
         <th>S No.</th>
         <th>Product Name</th>
         <th>Product Image</th>
         <th>Price</th>
         <th>Quantity</th>
         <th>Total Price</th>
         <th>Status</th>

      </tr>
       {% for order_var in fetch_order %}
      <tr class="text-center">
         <td>{{forloop.counter}}</td>
         <td>{{order_var.product.Product_name}}</td>
         <td>
            <img src="{{order_var.product.Product_image.url}}"
alt="" width=50 height=50>
         </td>
         <td>Rs.{{order_var.price}}</td>
         <td>{{order_var.quantity}}</td>

<td>Rs.{{order_var.price|order_total_price:order_var.quantity}}<
```

```
/td>
      <td>
        {% if order_var.status is False %}
            <span class="badge text-bg-
warning">Pending</span>
        {% else %}

            <span class="badge text-bg-
success">Completed</span>
        {% endif %}
      </td>
    </tr>

    {% endfor %}
    <tr class="text-center">
      <td colspan=5>
        <h3>Payable amount</h3>
      </td>
      <td>
        Rs.{{tp}}
      </td>
      <td>
         <div id="paypal-button-container"></div>
      </td>
    </tr>
  </table>
</div>
 <script src="https://www.paypal.com/sdk/js?client-
id=ATYcMDzCL3M_GX9cFHOfKzQSB6firqET4SxkdT8jMuOx
2Tdmukph16tCxn1lBVSBTFJzsbQhYGpRvNWK&currency=US
D"></script>
 <script>
    // Render the PayPal button into #paypal-button-container
    paypal.Buttons({

      // Set up the transaction
      createOrder: function(data, actions) {
        return actions.order.create({
          purchase_units: [{
            amount: {
              value: '{{tp}}'
```

```
                    }
                }]
            });
        },

        // Finalize the transaction
        onApprove: function(data, actions)


 {
            return actions.order.capture().then(function(orderData)
{
                // Successful capture! For demo purposes:
                console.log('Capture result', orderData,
JSON.stringify(orderData, null, 2));
                var transaction =
orderData.purchase_units[0].payments.captures[0];
                alert('Transaction '+ transaction.status + ': ' +
transaction.id + '\n\nSee console for all available details');

                // Replace the above to show a success message
within this page, e.g.
                // const element = document.getElementById('paypal-
button-container');
                // element.innerHTML = '';
                // element.innerHTML = '<h3>Thank you for your
payment!</h3>';
                // Or go to another URL:
actions.redirect('thank_you.html');
            });
        }


      }).render('#paypal-button-container');
   </script>
{% endblock content %}
```

Views.py:

```
from django.shortcuts import render,HttpResponse,redirect

# Create your views here.
from .models import *
from django.contrib.auth.hashers import check_password
,make_password
from django.contrib import messages

def index(request):
    if request.method == 'POST':
        product_id = request.POST.get('cartid')

        remove = request.POST.get('remove')

        cart_id = request.session.get('cart')

        if cart_id:
            quantity = cart_id.get(product_id)
            if quantity:
                if remove:
                    if quantity <= 1:
                            cart_id.pop(product_id)
                    else:
                        cart_id[product_id] = quantity - 1
                else:
                    cart_id[product_id] = quantity + 1
            else:
                cart_id[product_id] = 1
        else:
            cart_id = {}
            cart_id[product_id] = 1


        request.session['cart'] = cart_id


    category_obj = Category.objects.all()
    cat_id = request.GET.get('category_id')

    if cat_id:
        product_obj =
Product.objects.filter(Product_category=cat_id)
    else:
        product_obj = Product.objects.all()

    context={
        'category_obj': category_obj,
        'product_obj':product_obj
    }

    return render(request, 'home.html',context=context)




def contact(request):
    return render(request, 'contact.html')

def signup(request):
    if request.method == "POST":
        f_name = request.POST.get('fname')
        l_name = request.POST.get('lname')

        email = request.POST.get('email')
        password = request.POST.get('pwd')
        mobile = request.POST.get('mobile')
        gender = request.POST.get('gender')

        c_obj = Signup (
            first_name = f_name,
            last_name = l_name,
            email =email,
            password =make_password(password),
            mobile = mobile,
            gender  =gender,

        )
        c_obj.save()
        return redirect('home')
```

```
def login(request):
    if request.method == "POST":
        email_id = request.POST.get("emailid")
        password = request.POST.get("password")

        try:
            fetch_email = Signup.objects.get(email = email_id)
            if check_password(password,fetch_email.password):
                # return HttpResponse("Login successfull")
                request.session['name']= fetch_email.first_name
                request.session['customer_id']= fetch_email.id

                return redirect('home')
            else:
                error_message = "Wrong Password"



 except:
            error_message = "User Does not Exist."
    return render(request, 'base.html', {'error_message':
error_message})



def logout(request):
    request.session.clear()
    return redirect('home')

def cart_details(request):

    ids = list(request.session.get('cart').keys())

    cart_obj = Product.objects.filter(id__in=ids)
    return render(request,'cart.html',{'cart_obj':cart_obj})

def check_cart(request):
    if request.method == 'POST':
        address = request.POST.get('address')
        mobile = request.POST.get('mobile')
        customer_id = request.session.get('customer_id')
        messages.success(request,"order successfully created")


        if not customer_id:
            return HttpResponse("PLease Login........")

        cart = request.session.get('cart')
        product_details =
Product.objects.filter(id__in=list(cart.keys()))

        for pro in product_details:
            order_details = Order(
                address = address,
                mobile = mobile,
                customer = Signup(id=customer_id),
                product = pro,
                price = pro.Product_price,
                quantity = cart.get(str(pro.id))
        )


            order_details.save()
        return redirect('cart')
        # messages.success(request,"order successfully created")

        # return HttpResponse("order successfully created")


def order_details(request):

    customer_id = request.session.get('customer_id')
    fetch_order = Order.objects.filter(customer=customer_id)
    tp = 0
    for i in fetch_order:
        tp = tp + (i.price * i.quantity)

    context = {
        'fetch_order': fetch_order,
        'tp': tp
    }
    return render(request, 'order.html', context=context)

from rest_framework import routers, serializers, viewsets
from .serializations import SignUpSerializer
class UserViewSet(viewsets.ModelViewSet):
    queryset = Signup.objects.all()
    serializer_class = SignUpSerializer
```

Urls.py:

```
from django.contrib import admin
from django.urls import path,include
from . import views

from django.conf.urls.static import static
from django.conf import settings
from rest_framework import routers, serializers, viewsets
router = routers.DefaultRouter()
router.register(r'users', views.UserViewSet)


urlpatterns = [
    path("",views.index,name="home"),
    path("contact",views.contact,name="contact"),
    path("signup",views.signup,name="signup"),
    path("login",views.login,name="login"),
    path("logout",views.logout,name="logout"),
    path("cart",views.cart_details,name="cart"),
    path("checkout",views.check_cart, name="checkout"),
    path("order",views.order_details, name="order"),
     path('api/', include(router.urls)),

]
urlpatterns += static(settings.MEDIA_URL,
document_root=settings.MEDIA_ROOT)
```

Models.py

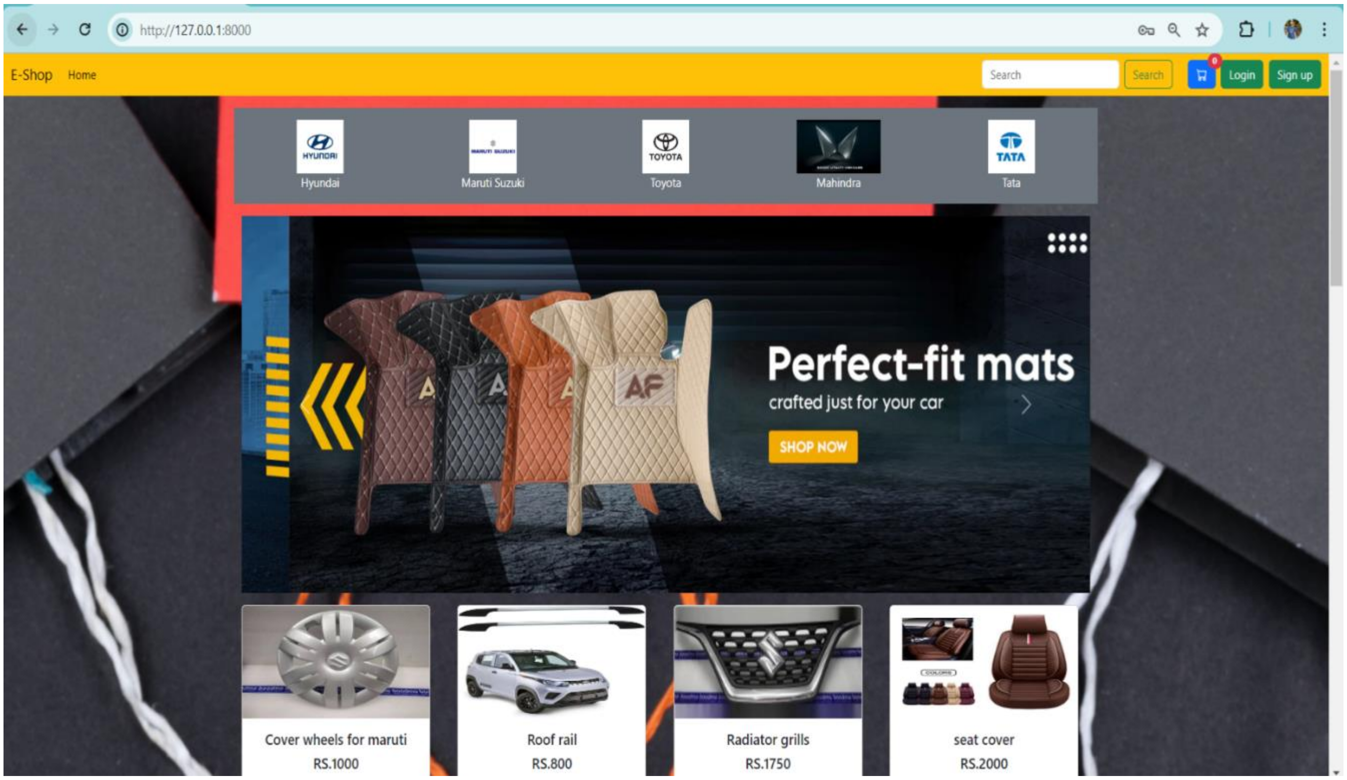
```
from django.db import models

# Create your models here.

class Signup(models.Model):
    first_name = models.CharField(max_length=50,null=True,blank=True)
    last_name = models.CharField(max_length=50,null=True,blank=True)
    email = models.CharField(max_length=50,null=True,blank=True)
    password = models.CharField(max_length=255,null=True,blank=True)

    mobile = models.BigIntegerField(default=0)
    gender = models.CharField(max_length=50,null=True,blank=True)



    def __str__(self):
        return self.first_name

class Category(models.Model):

Category_name=models.CharField(max_length=200,null=True,blank=Tr
ue)
    Category_image = models.ImageField(upload_to='upload/category/')

    def __str__(self):
        return self.Category_name

class Product(models.Model):
    Product_name =
models.CharField(max_length=100,null=True,blank=True)
    Product_desc =
models.CharField(max_length=200,null=True,blank=True)
    Product_price = models.IntegerField()
    Product_image = models.ImageField(upload_to='upload/product/')
    Product_category  = models.ForeignKey(Category,
on_delete=models.CASCADE)

    def __str__(self):
        return self.Product_name

class Order(models.Model):
    address = models.CharField(max_length=200, blank=True, null=True)
    mobile = models.BigIntegerField()
    customer = models.ForeignKey(Signup,
on_delete=models.CASCADE)
    product = models.ForeignKey(Product, on_delete=models.CASCADE
)
    price = models.BigIntegerField()
    quantity = models.IntegerField()
    status = models.BooleanField(default=False)

    def __str__(self):
        return self.product.Product_name
```

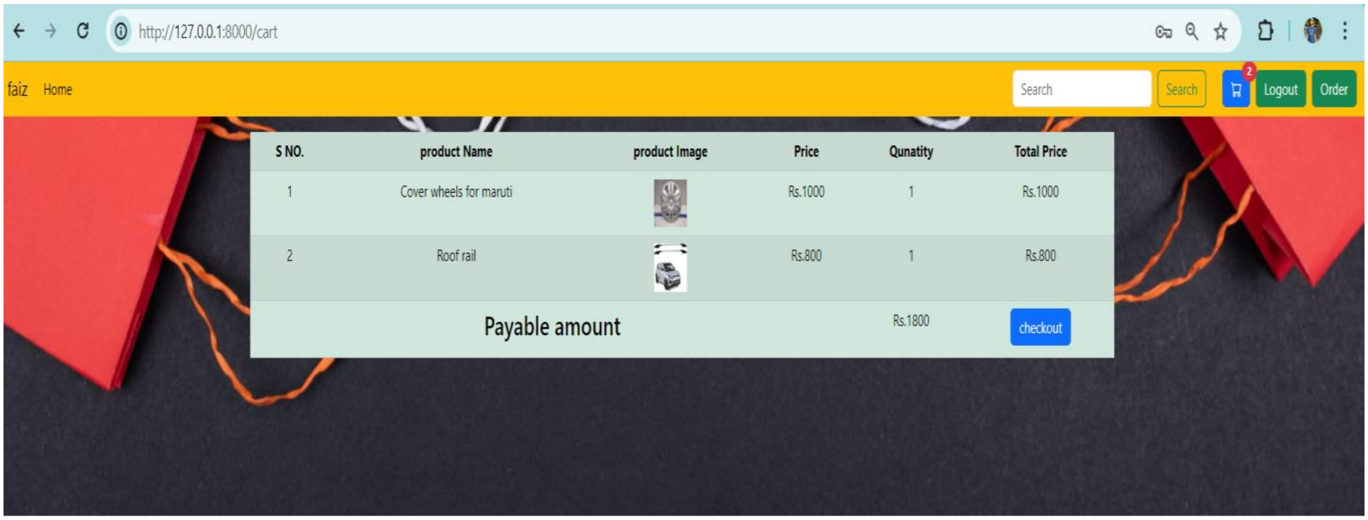
Admin.py:

```
from django.contrib import admin

# Register your models here.

from .models import *

admin.site.register(Signup)
admin.site.register(Category)
admin.site.register(Product)
admin.site.register(Order)
```

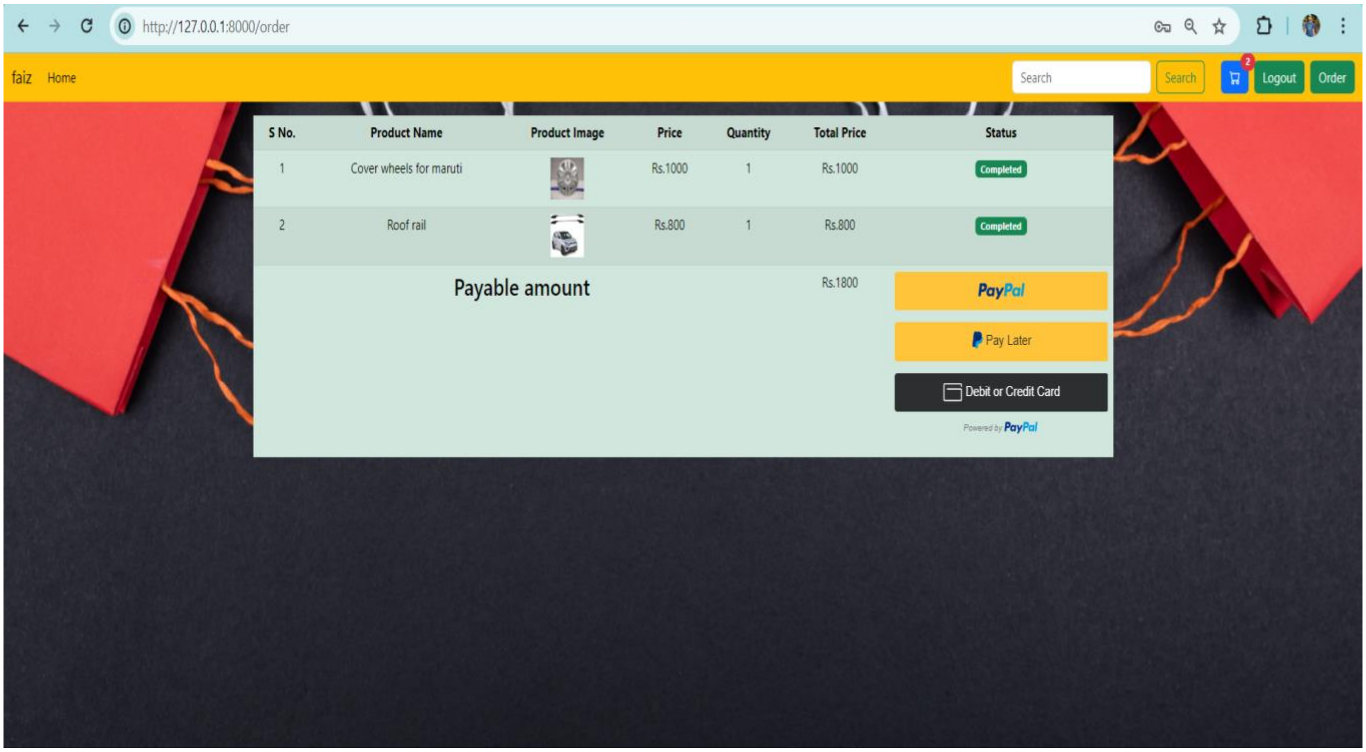
6.4 SCREENSHOTS

h

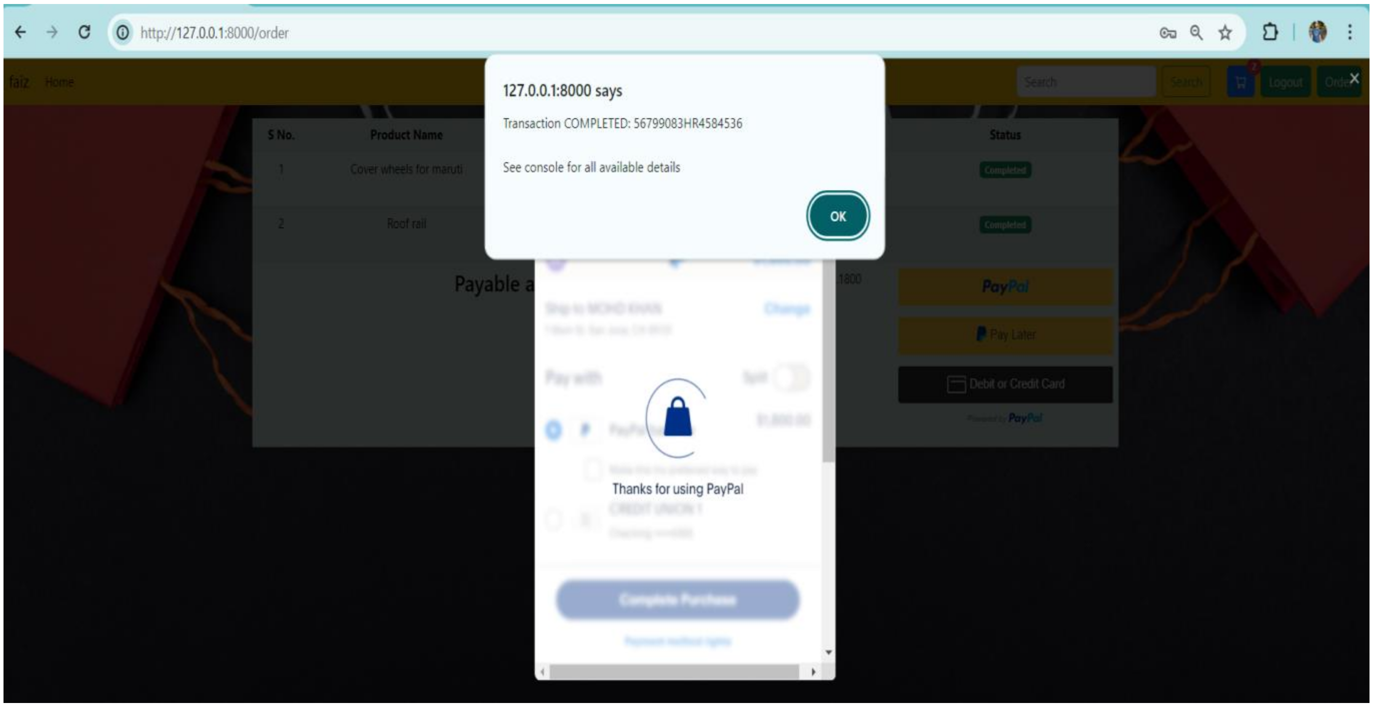




S NO.	product Name	product Image	Price	Qunatity	Total Price
1	Cover wheels for maruti		Rs.1000	1	Rs.1000
2	Roof rail		Rs.800	1	Rs.800
	Payable amount			Rs.1800	checkout



S No.	Product Name	Product Image	Price	Quantity	Total Price	Status
1	Cover wheels for maruti		Rs.1000	1	Rs.1000	Completed
2	Roof rail		Rs.800	1	Rs.800	Completed
	Payable amount				Rs.1800	

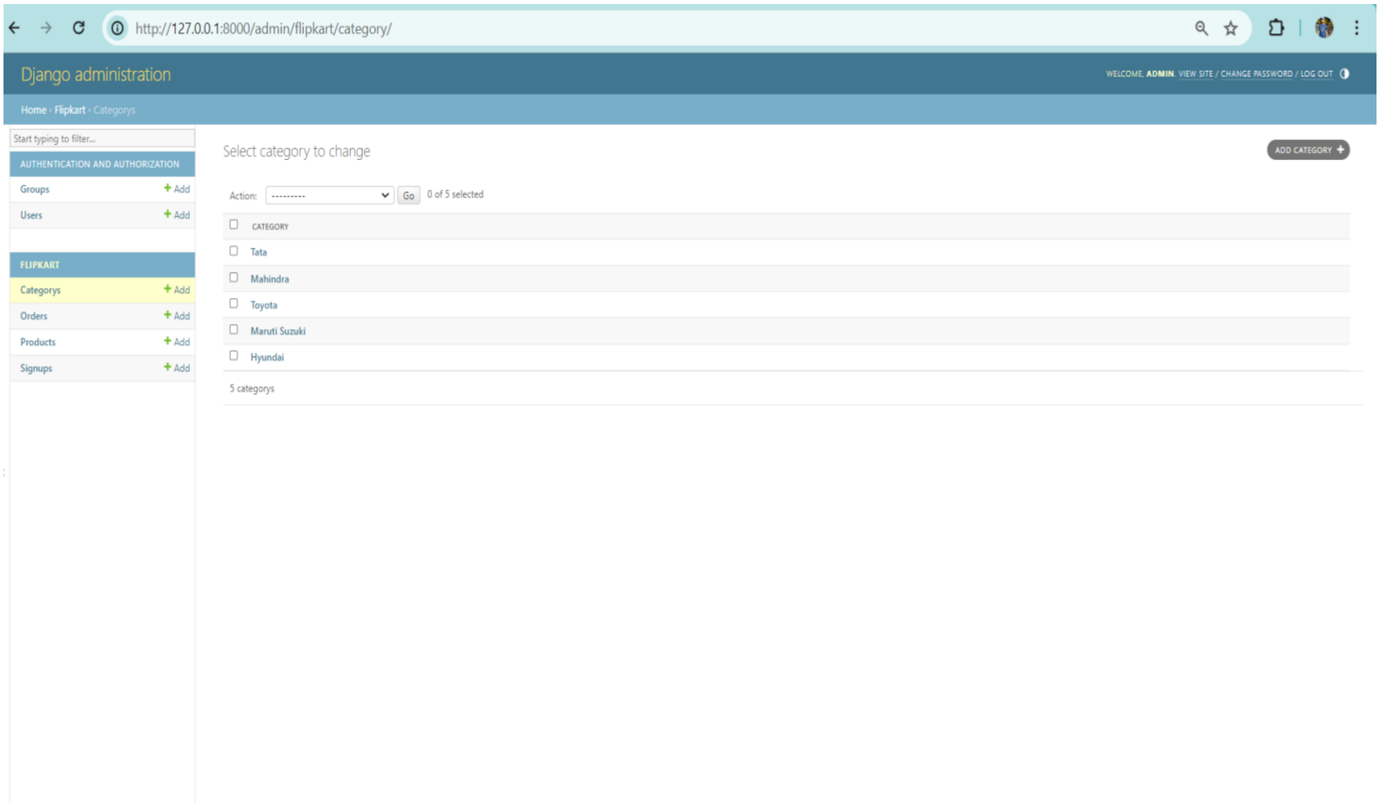


127.0.0.1:8000 says

Transaction COMPLETED: 56799083HR4584536

See console for all available details

Thanks for using PayPal


http://127.0.0.1:8000/admin/flipkart/category/
Django administration
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Flipkart › Categorys
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
FLIPKART
Categorys + Add
Orders + Add
Products + Add
Signups + Add
ADD CATEGORY +
Action: --------- Go

Select category to change

0 of 5 selected

□	CATEGORY
□	Tata
□	Mahindra
□	Toyota
□	Maruti Suzuki
□	Hyundai

5 categorys

6.5 MAINTENANCE

The maintenance phase involves making changes to hardware, software, and documentation to support its operational effectiveness. It includes making changes to improve a system's performance, correct problems, enhance security, or address user requirements. To ensure modifications do not disrupt operations or degrade a system's performance or security, organizations should establish appropriate change management standards and procedures.

Routine changes are not as complex as major modifications and can usually be implemented in the normal course of business. Routine change controls should include procedures for requesting, evaluating, approving, testing, installing, and documenting website modifications. Maintaining accurate, up-to-date hardware and software inventories is a critical part of all change management processes. Management should carefully document all modifications to ensure accurate system inventories. Management should coordinate all technology related changes through an oversight committee and assign an appropriate party responsibility for administering software patch management programs. Quality assurance, security, audit, regulatory compliance, network, and end-user personnel should be appropriately included in change management processes. Risk and security review should be done whenever a system modification is implemented to ensure controls remain in place.

6.6 FUTURE SCOPE & FUTURE ENHANCEMENT

PROJECT NAME:

SHOP HOUSE

1. Shop House would help each and every person to find any automobile parts via our website and get it at home it will save their time.

2. It would provide huge collection of automobile parts of all fields.

3. We will be providing some special offers in online.

6.7 CONCLUSION

We have successfully implemented the site “Shop House”. With the help of various links and tools, we have been able to provide a site which will be live soon and running on the web. We have been successful in our attempt to take care of the needs of both the user as well as the administrator. Finally we hope that this will go a long way in popularizing.

6.8 REFERENCS

1. www.javatutpoint.com
2. www.w3schools.com
3. www.getbootstrap.com
4. www.codeigniter.com
5. www.stackoverflow.com
6. www.fontawesome.io
7. Learn HTML and CSS faster(Mark Myers)
8. Wikipedia

Published 1st Research Papers hard copy along with certificates

iJRASET

International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 Issue: XII Month of publication: December 2023

DOI: https://doi.org/10.22214/ijraset.2023.57390

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue XII Dec 2023- Available at www.ijraset.com

Shope House

Mohd Amaan Khan[1], Sahil Khan[2], Faiz Mohammad[3]
[1, 2, 3] *Students, Computer Science and Engineering, Babu Banarasi Das Northern India Institute of Technology*

Mr. Waseem Ahmad
Assistant Professor, Computer Science and Engineering, Babu Banarasi Das Northern India Institute of Technology

Abstract: A Shop House System for Web Applications in the Automotive Industry: Enabling Seamless Collaboration between Buyers and Automotive Companies

In the rapidly evolving landscape of the Automotive industry, the integration of web applications has become instrumental in facilitating efficient communication and transactions between buyers and Automotive companies. This research paper explores the design, development, and implementation of a robust Shop House system tailored for a web application in the Automotive sector. The proposed Automotive system serves as a pivotal tool for both buyers and Automotive entities, fostering seamless interactions, streamlined processes, and an enhanced user experience.

Keywords: Automotive industry, Transactions, seamless interaction, enhanced user experience,Digital transformation

I. INTRODUCTION

The intricacies of the proposed Shop house system lie in its ability to intricately weave together the needs of both buyers and Automotive companies within the web application framework. For buyers, the system facilitates a user-friendly interface for secure prescription uploads, granting access to comprehensive product information and enabling confident, well-informed purchasing decisions. The intuitive design ensures a seamless experience, where users can navigate, inquire, and transact with ease.

On the Automotive company side, the Shop House system provides a dynamic platform for showcasing product portfolios, coupled with real-time inventory management and automated order processing. This functionality not only streamlines internal processes but also enhances responsiveness to market demands. The system's capability to harness advanced data analytics becomes pivotal, offering personalized recommendations to buyers and empowering Automotive companies with valuable insights into market trends. This analytical prowess ensures that both buyers and Automotive entities can adapt swiftly to the ever-changing landscape of the Automotive industry.

Moreover, the Shop House system operates as a central hub for communication and collaboration, fostering transparency and trust throughout the supply chain. Its architecture is crafted to accommodate the specific needs of the Automotive sector, incorporating stringent security measures to protect sensitive data and ensuring compliance with regulatory standards. This holistic approach to functionality and security positions the Shop House system as a transformative force, driving efficiency, innovation, and enhanced relationships within the web application ecosystem of the Automotive industry.

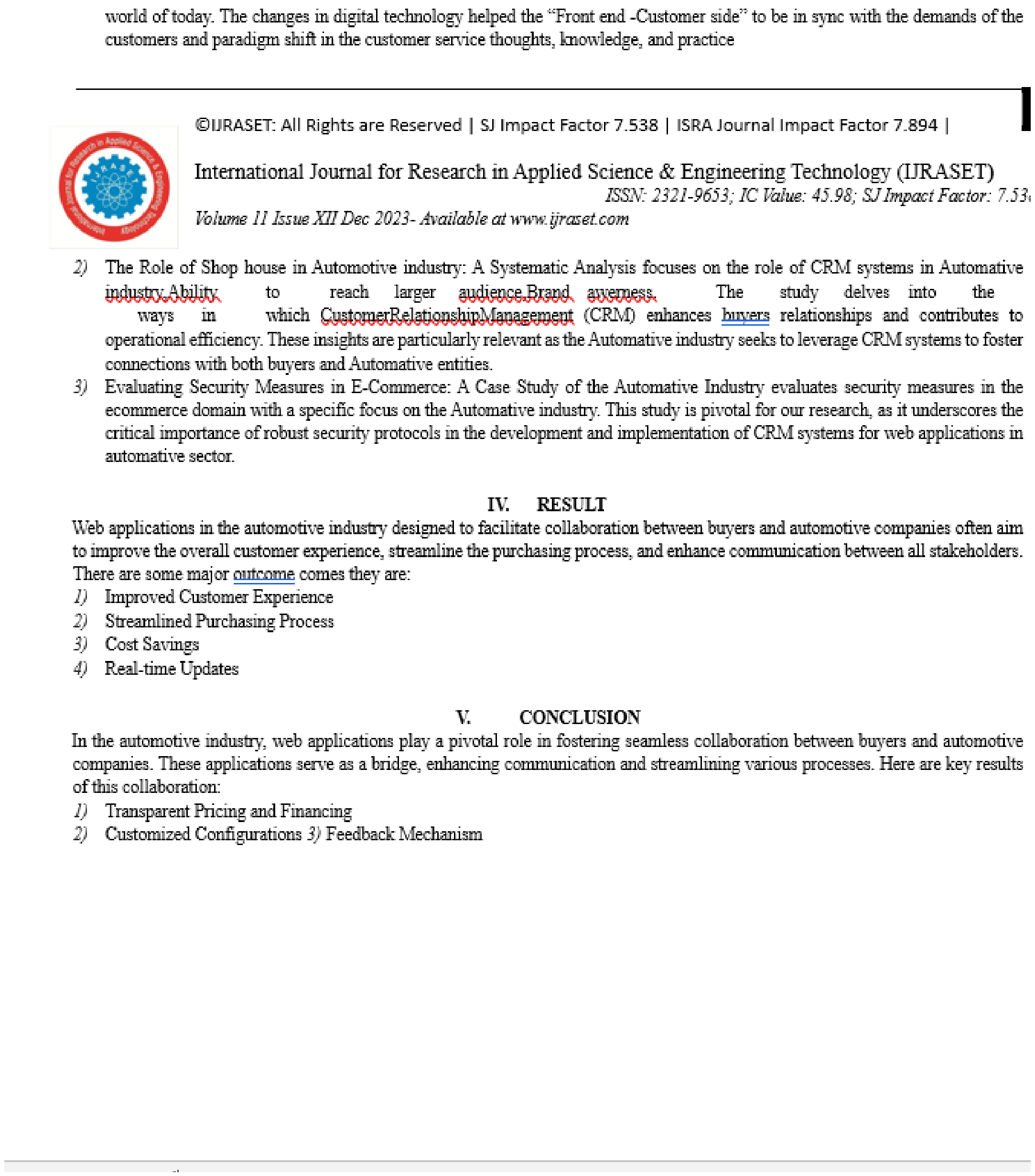
Key Words: Web Application Integration, Advanced Data Analytics, Supply Chain Transparency, Operational Optimization

II. PROBLEM STATEMENT

In navigating the digital evolution of the Automotive sector, a key puzzle piece is missing – a dedicated system that seamlessly connects buyers and Automotive companies through web applications. Without a tailored Shop House solution, challenges arise in managing from a Suspicious Seller or website, updating inventory in real-time, and establishing personalized connections. This gap impedes the industry's pursuit of transparency, smooth operations, and inventive approaches. This study endeavors to fill this void by introducing a specialized Shop house solution, simplifying the digital Automotive supply chain for enhanced collaboration and operational efficiency between buyers and Automotive entities.

III. LITERATURE REVIEW

1) Digital Transformation in Automotive Retail: Digital transformation and its relationship & impact of front end (Customer), especially on customer experience and customer satisfaction has been a focus of study for last three decades. What has started purely as a "After sales service" initially changed to service dominant logic, Customer relationship management and customer experience with evolution in thoughts accompanied by deep leaps in digital technology from computers of 1960's to totally digital



world of today. The changes in digital technology helped the "Front end -Customer side" to be in sync with the demands of the customers and paradigm shift in the customer service thoughts, knowledge, and practice

2) The Role of Shop house in Automotive industry: A Systematic Analysis focuses on the role of CRM systems in Automative industry.Ability to reach larger audience.Brand awerness. The study delves into the ways in which CustomerRelationshipManagement (CRM) enhances buyers relationships and contributes to operational efficiency. These insights are particularly relevant as the Automative industry seeks to leverage CRM systems to foster connections with both buyers and Automative entities.

3) Evaluating Security Measures in E-Commerce: A Case Study of the Automative Industry evaluates security measures in the ecommerce domain with a specific focus on the Automative industry. This study is pivotal for our research, as it underscores the critical importance of robust security protocols in the development and implementation of CRM systems for web applications in automative sector.

IV. RESULT

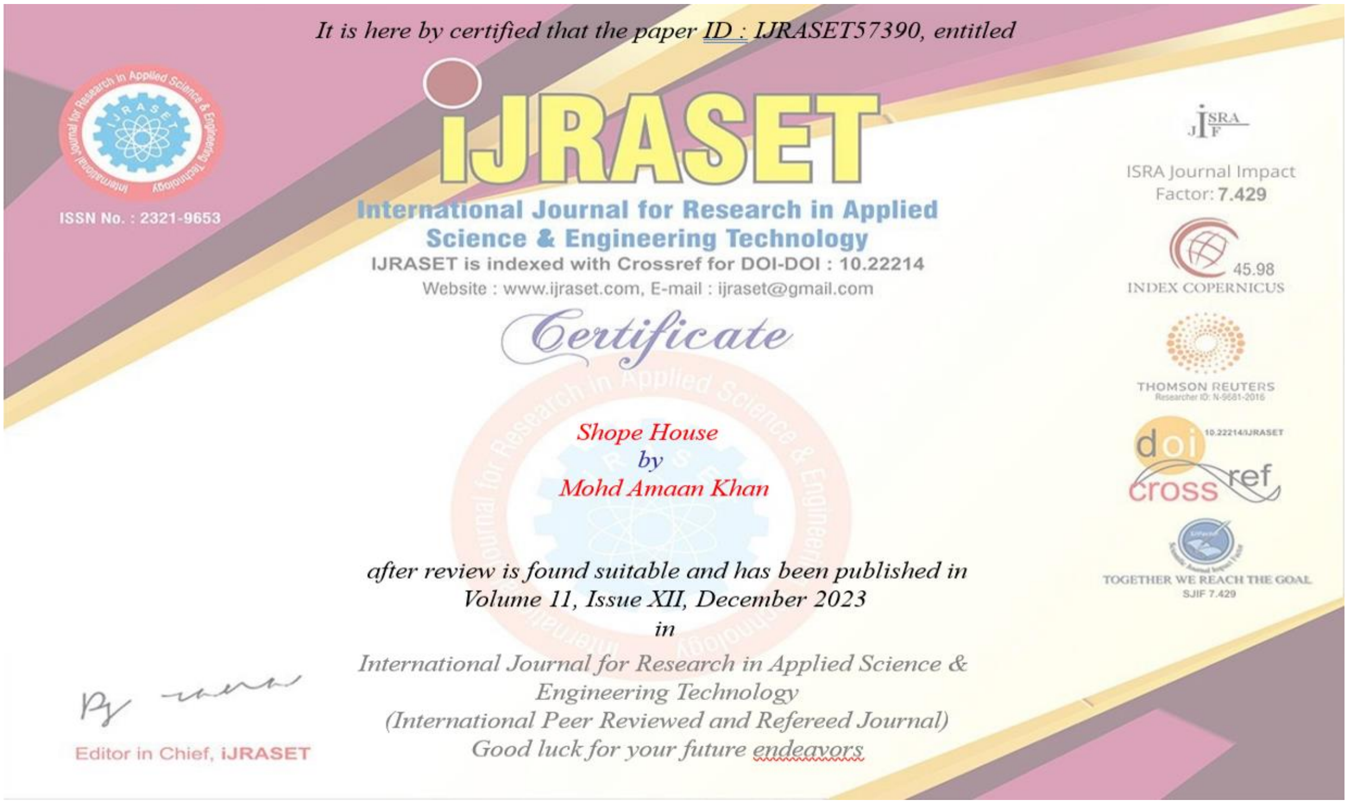
Web applications in the automotive industry designed to facilitate collaboration between buyers and automotive companies often aim to improve the overall customer experience, streamline the purchasing process, and enhance communication between all stakeholders. There are some major outcome comes they are:

1) Improved Customer Experience
2) Streamlined Purchasing Process
3) Cost Savings
4) Real-time Updates

V. CONCLUSION

In the automotive industry, web applications play a pivotal role in fostering seamless collaboration between buyers and automotive companies. These applications serve as a bridge, enhancing communication and streamlining various processes. Here are key results of this collaboration:

1) Transparent Pricing and Financing
2) Customized Configurations *3)* Feedback Mechanism



It is here by certified that the paper ID : IJRASET57390, entitled

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

Certificate

Shope House
by
Mohd Amaan Khan

after review is found suitable and has been published in
Volume 11, Issue XII, December 2023
in
International Journal for Research in Applied Science &
Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET

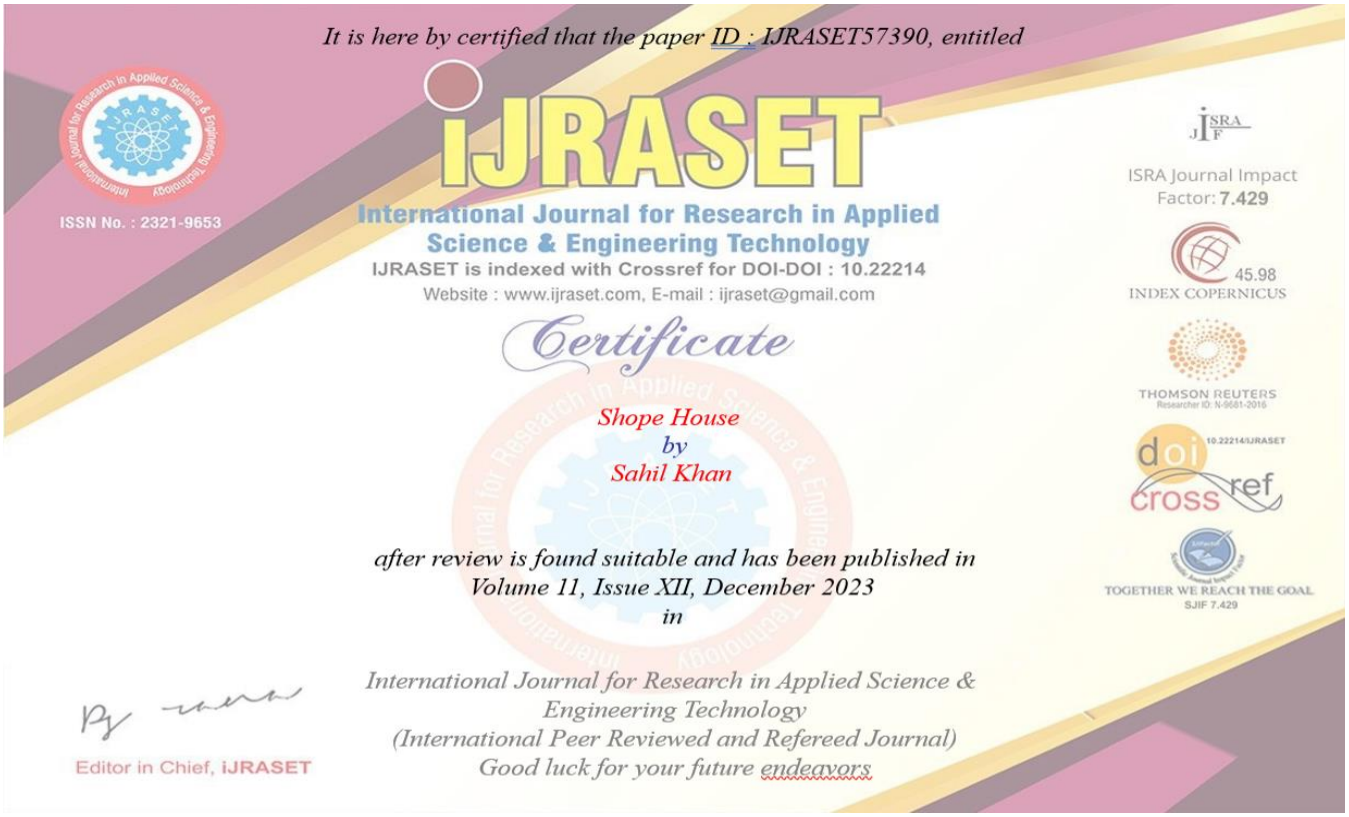
ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429



It is here by certified that the paper ID : IJRASET57390, entitled

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

Certificate

Shope House
by
Sahil Khan

after review is found suitable and has been published in
Volume 11, Issue XII, December 2023
in
International Journal for Research in Applied Science &
Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET

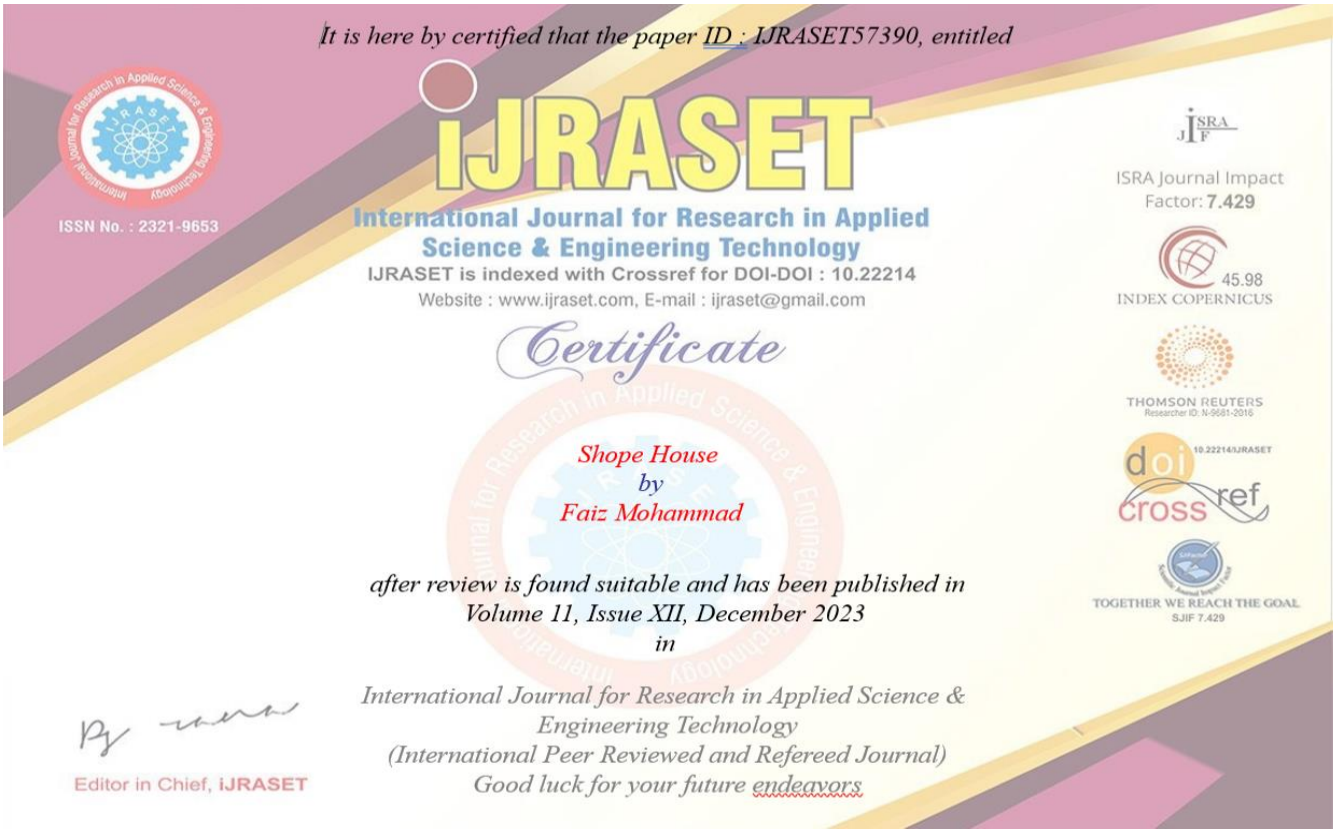
ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429



It is here by certified that the paper ID : IJRASET57390, entitled

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISSN No. : 2321-9653

Certificate

Shope House

by

Faiz Mohammad

after review is found suitable and has been published in

Volume 11, Issue XII, December 2023

in

International Journal for Research in Applied Science & Engineering Technology

(International Peer Reviewed and Refereed Journal)

Good luck for your future endeavors

Editor in Chief, iJRASET

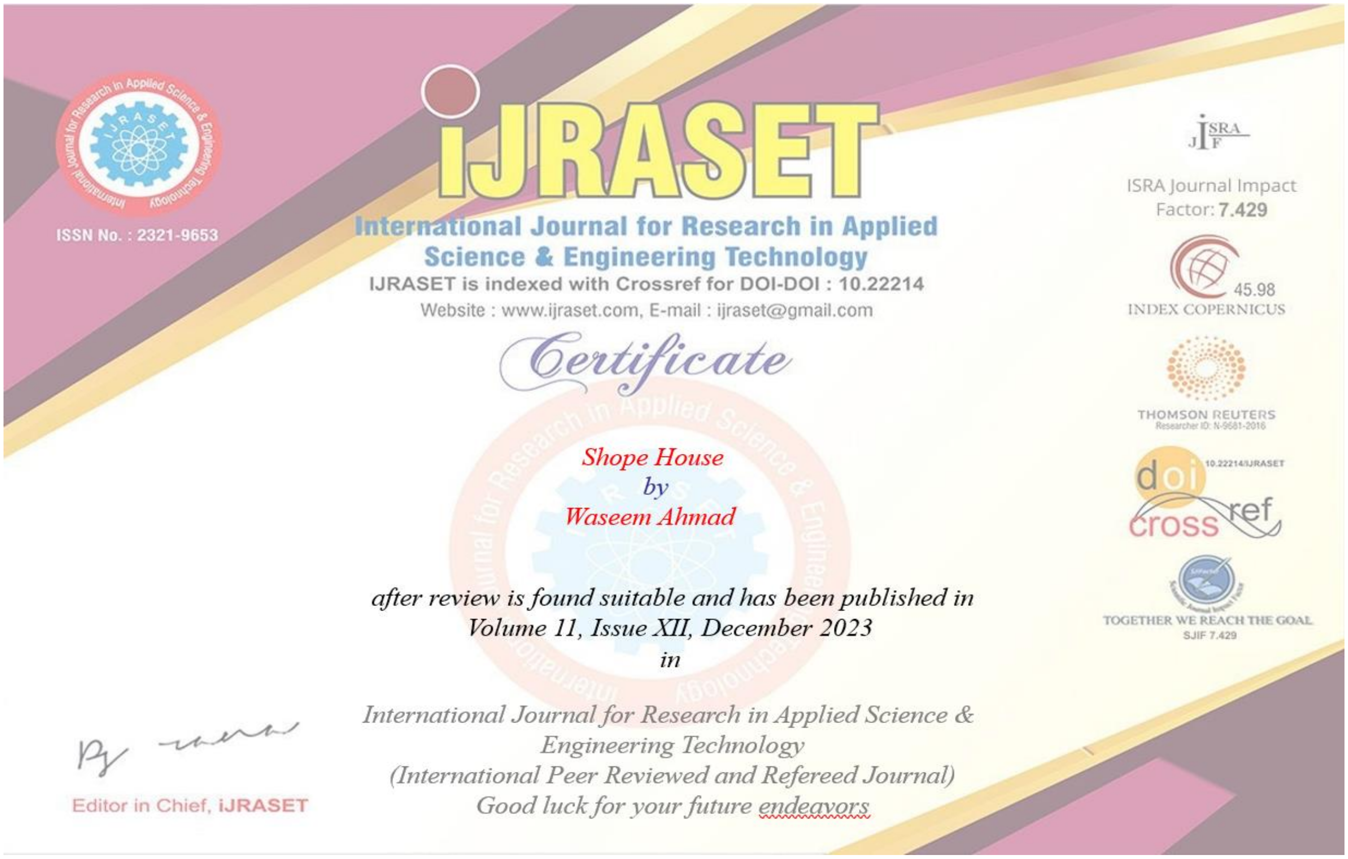
ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429



iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISSN No. : 2321-9653

Certificate

Shope House

by

Waseem Ahmad

after review is found suitable and has been published in

Volume 11, Issue XII, December 2023

in

International Journal for Research in Applied Science & Engineering Technology

(International Peer Reviewed and Refereed Journal)

Good luck for your future endeavors

Editor in Chief, iJRASET

ISRA Journal Impact Factor: 7.429

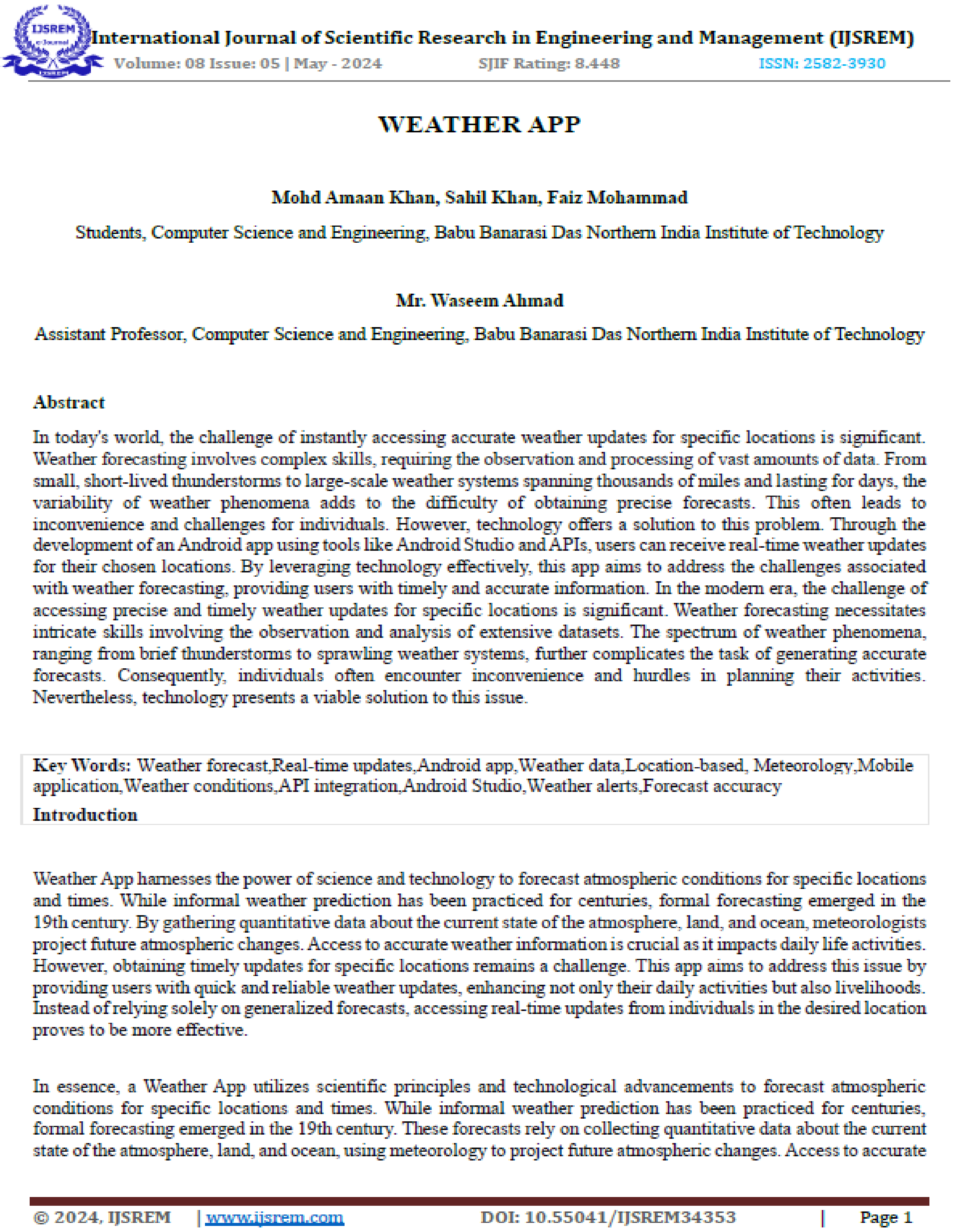
45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

Published 2nd Research Papers hard copy along with certificates



International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

WEATHER APP

Mohd Amaan Khan, Sahil Khan, Faiz Mohammad

Students, Computer Science and Engineering, Babu Banarasi Das Northern India Institute of Technology

Mr. Waseem Ahmad

Assistant Professor, Computer Science and Engineering, Babu Banarasi Das Northern India Institute of Technology

Abstract

In today's world, the challenge of instantly accessing accurate weather updates for specific locations is significant. Weather forecasting involves complex skills, requiring the observation and processing of vast amounts of data. From small, short-lived thunderstorms to large-scale weather systems spanning thousands of miles and lasting for days, the variability of weather phenomena adds to the difficulty of obtaining precise forecasts. This often leads to inconvenience and challenges for individuals. However, technology offers a solution to this problem. Through the development of an Android app using tools like Android Studio and APIs, users can receive real-time weather updates for their chosen locations. By leveraging technology effectively, this app aims to address the challenges associated with weather forecasting, providing users with timely and accurate information. In the modern era, the challenge of accessing precise and timely weather updates for specific locations is significant. Weather forecasting necessitates intricate skills involving the observation and analysis of extensive datasets. The spectrum of weather phenomena, ranging from brief thunderstorms to sprawling weather systems, further complicates the task of generating accurate forecasts. Consequently, individuals often encounter inconvenience and hurdles in planning their activities. Nevertheless, technology presents a viable solution to this issue.

Key Words: Weather forecast,Real-time updates,Android app,Weather data,Location-based, Meteorology,Mobile application,Weather conditions,API integration,Android Studio,Weather alerts,Forecast accuracy

Introduction

Weather App harnesses the power of science and technology to forecast atmospheric conditions for specific locations and times. While informal weather prediction has been practiced for centuries, formal forecasting emerged in the 19th century. By gathering quantitative data about the current state of the atmosphere, land, and ocean, meteorologists project future atmospheric changes. Access to accurate weather information is crucial as it impacts daily life activities. However, obtaining timely updates for specific locations remains a challenge. This app aims to address this issue by providing users with quick and reliable weather updates, enhancing not only their daily activities but also livelihoods. Instead of relying solely on generalized forecasts, accessing real-time updates from individuals in the desired location proves to be more effective.

In essence, a Weather App utilizes scientific principles and technological advancements to forecast atmospheric conditions for specific locations and times. While informal weather prediction has been practiced for centuries, formal forecasting emerged in the 19th century. These forecasts rely on collecting quantitative data about the current state of the atmosphere, land, and ocean, using meteorology to project future atmospheric changes. Access to accurate weather information is vital as it impacts daily activities. Rapid updates for specific locations enhance efficiency in managing activities and livelihoods. Presently, a significant challenge lies in obtaining real-time weather updates for particular places. Thus, consulting individuals currently in those locations proves more reliable than relying solely on general weather forecasts.

In summary, a Weather App employs scientific principles and technological advancements to predict atmospheric conditions for specific locations and times. While informal weather forecasting has historical roots, formal forecasting emerged in the 19th century. These predictions rely on gathering quantitative data about the current state of the atmosphere, land, and ocean, using meteorology to forecast future atmospheric changes.

Key Words: Weather App,Scientific principles,Technological advancements,Forecasting atmospheric conditions,Informal weather prediction,Formal forecasting,Quantitative data,Atmosphere, land, and ocean

Literature Review

The Weather App, as its name suggests, is an innovative system designed to swiftly provide accurate and reliable weather data to users, particularly tourists. This Android application utilizes web design languages for its front-end interface and integrates with APIs to enhance functionality. Serving as a weather forecaster, the app delivers precise outputs based on user inputs. It boasts high reliability, leveraging the foursquare API for accurate location data and ensuring precise weather conditions. The system aims to offer users timely insights into weather conditions, ensuring their comfort and convenience during their visits. Users can select various destinations, such as parks, beaches, monuments, or dining spots, with the system offering flexibility to switch between locations effortlessly. By accessing the app's portal, users can easily search for their desired destinations and obtain detailed weather forecasts for specific dates or upcoming days.

Need of Application

The Weather App serves a vital purpose in several aspects:

- Weather forecasting aids in predicting future climate changes by utilizing latitude to assess the likelihood of snow and hail reaching the surface. It also helps determine the thermal energy received from the sun in specific regions. Climatology, the study of climates over time, relies on these variables and averages of short-term and long-term weather conditions. Distinct from meteorology, climatology offers various avenues for research and contributes to the development of efficient environmental activities.
- Agriculture heavily depends on seasons and nature. Temperature is particularly crucial for the cultivation of fruits, vegetables, and pulses. Historically, farmers relied on estimations due to the lack of accurate weather forecasts, leading to potential losses. However, with technological advancements and specialized weather forecasting mechanisms, farmers can now access forecasts directly on their smartphones. While education in this field remains essential, the widespread understanding of fundamental concepts among farmers facilitates the utilization of these features, enhancing agricultural practices.

CONCLUSION

In conclusion, it is evident that a weather application holds significant value for users, especially considering the widespread use of smartphones and the convenience they offer. While weather apps can be developed for various platforms, targeting Android would provide access to a larger user base due to its extensive popularity. Moreover, Android apps are typically more cost-effective and accessible, enhancing user reach. By delivering real-time weather updates, forecasts, and relevant data, such an application empowers users to make informed decisions about their daily activities. Additionally, it can offer users access to statistical information and trends, enabling them to track weather patterns over time and make well-informed choices based on data. In essence, a weather application serves as a valuable tool for users to stay informed about weather conditions and make informed decisions accordingly.

FUTURE SCOPE

The future outlook for weather applications appears promising, driven by the growing demand for accurate and real-time weather information. Advancements in data collection, analysis techniques, and algorithmic sophistication are expected to enhance the accuracy of forecasts, benefiting individuals and organizations alike by enabling informed decision-making. Personalization is poised to play a pivotal role in the evolution of weather apps, with tailored forecasts and alerts based on user location, preferences, and behavior. This customization will render weather information more pertinent and actionable for users. Additionally, there is a trend towards making weather applications more intuitive and user-friendly, employing visualizations and other tools to simplify the comprehension of complex weather data, further enhancing the user experience.

REFERENCES

[1] https://www.vedantu.com/geography/weather-forecasting

[2] https://en.wikipedia.org/wiki/Android_Studio

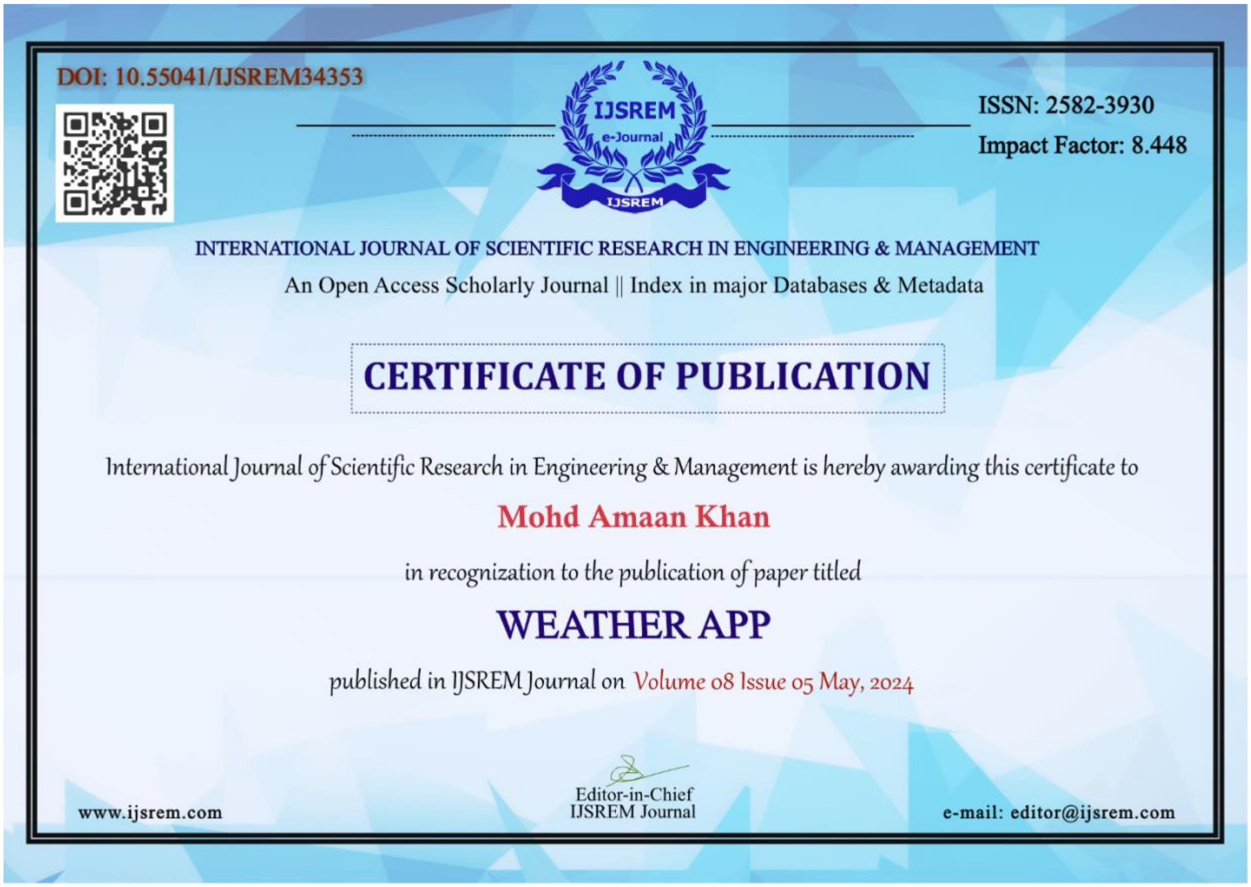
[3] OpenWeatherMap - Provides weather data API and sample codes for various platforms. https://openweathermap.org/

[4] Android Developers - Offers comprehensive documentation and tutorials for Android app development, including weather applications. https://developer.android.com/docs

[5] W3Schools: This website is one of the most popular and comprehensive resources for learning HTML and CSS. It provides tutorials, examples, and references for all aspects of web development, including HTML, CSS, and JavaScript. https://www.w3schools.com/

[6] MDN Web Docs : This website is a project by Mozilla, and it provides comprehensive documentation and tutorials for web development technologies, including HTML and CSS.

[7] https://developer.mozilla.org/en-US/docs/Web



DOI: 10.55041/IJSREM34353

ISSN: 2582-3930

Impact Factor: 8.448

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Mohd Amaan Khan

in recognization to the publication of paper titled

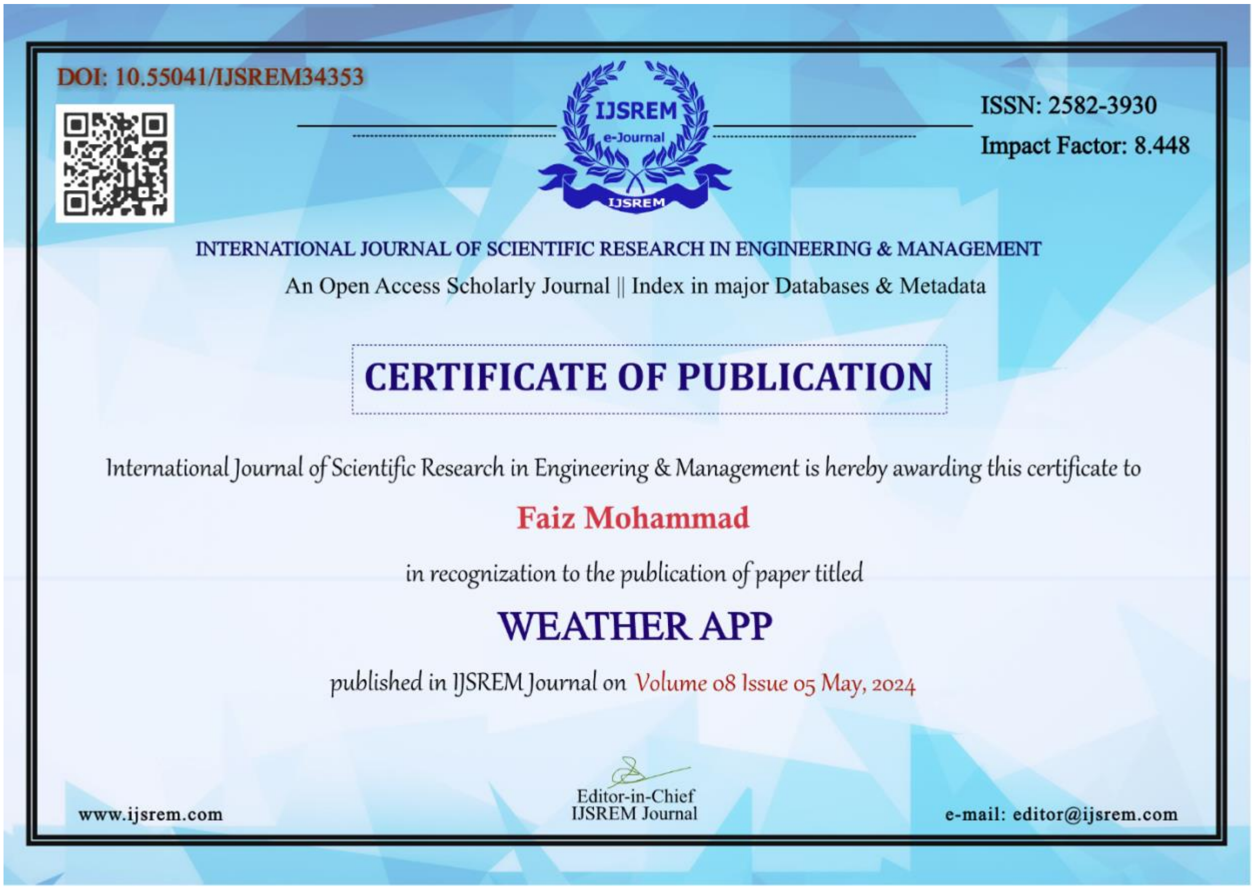
WEATHER APP

published in IJSREM Journal on Volume 08 Issue 05 May, 2024

www.ijsrem.com

Editor-in-Chief
IJSREM Journal

e-mail: editor@ijsrem.com



DOI: 10.55041/IJSREM34353

ISSN: 2582-3930

Impact Factor: 8.448

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Faiz Mohammad

in recognization to the publication of paper titled

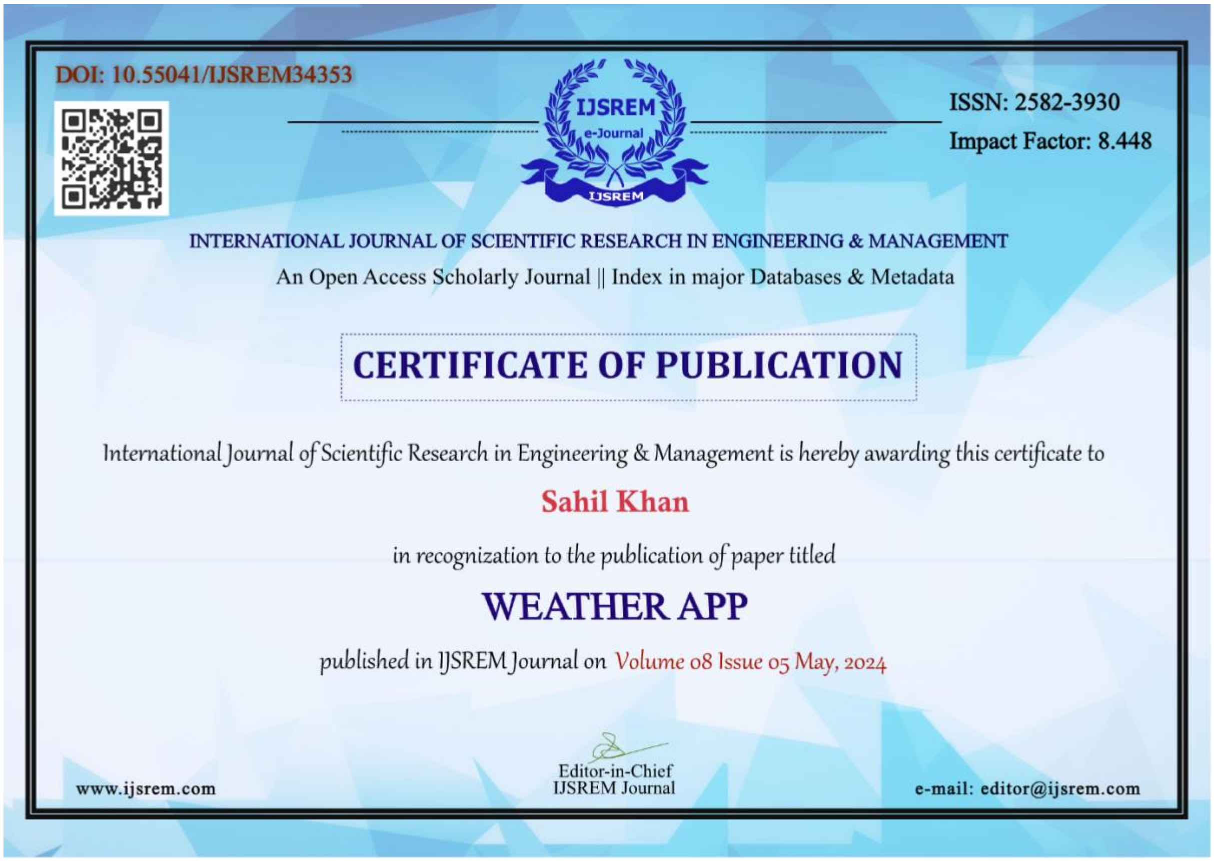
WEATHER APP

published in IJSREM Journal on Volume 08 Issue 05 May, 2024

www.ijsrem.com

Editor-in-Chief
IJSREM Journal

e-mail: editor@ijsrem.com



DOI: 10.55041/IJSREM34353

ISSN: 2582-3930

Impact Factor: 8.448

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Sahil Khan

in recognization to the publication of paper titled

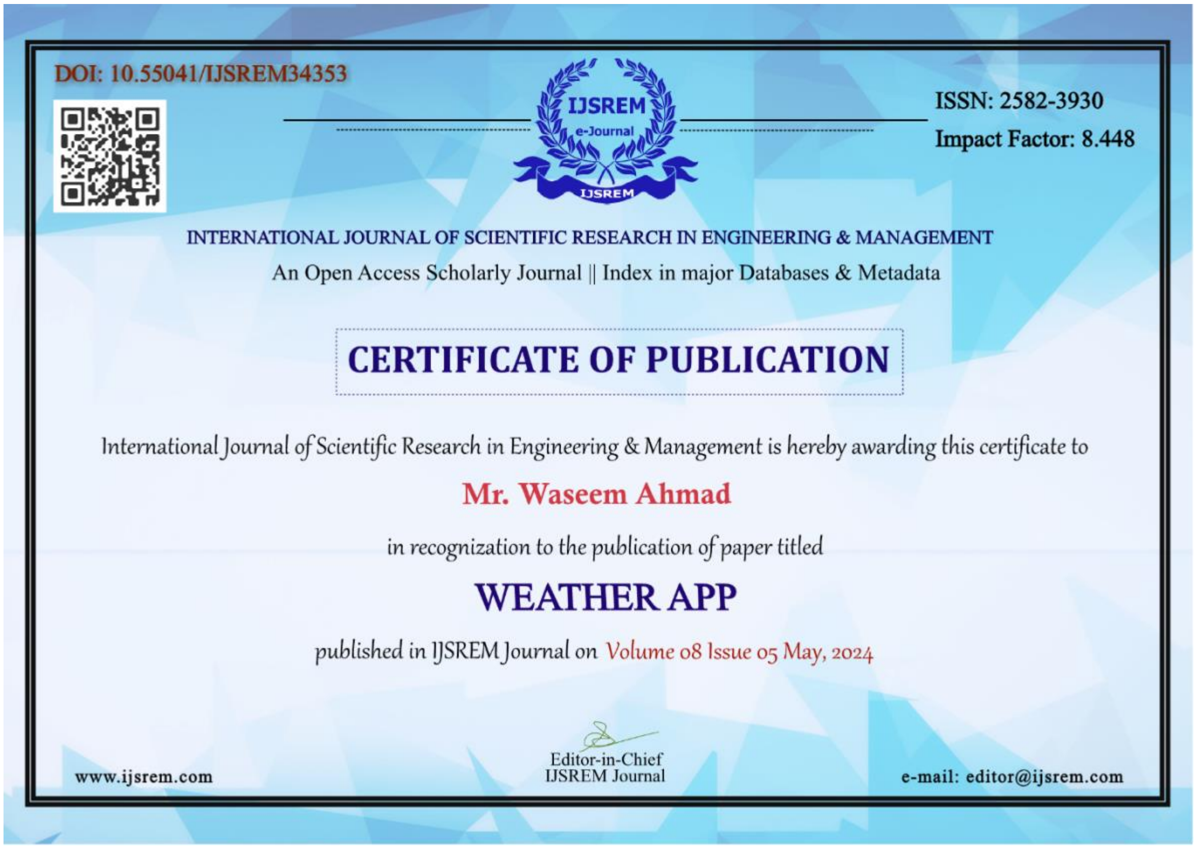
WEATHER APP

published in IJSREM Journal on Volume 08 Issue 05 May, 2024

Editor-in-Chief
IJSREM Journal

www.ijsrem.com

e-mail: editor@ijsrem.com



DOI: 10.55041/IJSREM34353

ISSN: 2582-3930

Impact Factor: 8.448

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

Mr. Waseem Ahmad

in recognization to the publication of paper titled

WEATHER APP

published in IJSREM Journal on Volume 08 Issue 05 May, 2024

Editor-in-Chief
IJSREM Journal

www.ijsrem.com

e-mail: editor@ijsrem.com