**Six weeks Industrial Training Project Report**

**On**

**“Veggie Delight”**

Submitted in the partial fulfilment of the requirement for the award of degree of

Bachelor of Technology

In

Computer Science and Engineering

Batch

(2020 - 2024)

A logo with a flower and a star

Description automatically generated

**Submitted to:                                                                   Submitted by:**

Dr. Rahul Hans                                                                         Hritik Kumar

Head of the department (CSE)                                   12000107

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**DAV UNIVERSITY**

**JALANDHAR-PATHANKOT NATIONAL HIGHWAY, NH 44,**

**SARMASTPUR, PUNJAB**

**144012**

# ACKNOWLEDGEMENT

I express my gratitude to all those who helped us in various stages of the development of this project. First, I would like to express my sincere gratitude indebtedness to Dr. Manoj Kumar (Vice-Chancellor), Dr Rahul Hans (Head of the department), Dr. Naveen Bilandi (Training and Placement Head, Department of Computer Science and Engineering) DAV University for allowing me to undergo the summer training of 45 days at O7 SERVICES.

I am also thankful to all faculty members of Department of Computer Science and Engineering, for their true help, inspiration and for helping me for the preparation of the final report and presentation.

Finally, I pay my sincere thanks and gratitude to all the Staff Members of O7 SERVICES for their support and for making our training valuable and fruitful.

**DECLARATION**

I, Hritik Kumar, hereby declare that the work which is being presented in this project/training titled “Veggie Delights” by me, in partial fulfilment of the requirements for the award of Bachelor of Technology (B. Tech) Degree in “Computer Science and Engineering” is an authentic record of my own work carried out under the guidance of Ms. Damanpreet.

To the best of my knowledge, the matter embodied in this report has not been submitted to any other University/ Institute for the award of any degree or diploma.

Hritik Kumar

12000107

**CERTIFICATE**



**COMPANY PROFILE**

****

O7 Services is an **ISO 9001:2015** **and Govt. Authorized** Company. They deal in Web Development, Mobile Application Development, Custom Software Development, UI/ UX Designing, Hosting services, Digital Marketing, Registration of Domain Names with the various latest extensions, AMC & MMC Services, Bulk sms and voice calls. They provide the most advanced IT solutions, supporting a full business cycle: preliminary consulting, system development, deployment, quality assurance and 24×7 support. With a rich experience of over 8 successful years, O7 Services tends to build long-lasting strategic partnerships with their clients to ensure affordable prices, timely delivery and measurable business results. Their Head Office is in Jalandhar and the Branch Office is in Hoshiarpur. Some of the products developed by O7 Services are- Vehicle Tracking System, Invoice Software, School Management System, Hospital Management system, Parents- Teacher Communication App, Fee Management system, Task Management System, Online Food Ordering App, Security App, Admission system, Inventory Software, Car Servicing App etc.

Apart from this, O7 Services also provides 6 Weeks/ 6 Months Industrial Training, Project Based Training, Corporate Training and Job Oriented Courses Training to the students on all major IT Trends like-

Full Stack Development (MEAN/ MERN), Flutter, Kotlin Android, Swift UI (iOS), Firebase, Python, Angular, React Js, Vue Js, Node Js, ASP.NET, .NET Core, PHP, Laravel, CodeIgniter, Software Testing, Cloud Computing, Blockchain, DevOps, Data Science, Artificial Intelligence, Machine Learning, IoT, UI/ UX Designing, Digital Marketing, WordPress, Linux, CCNA, CCNP, CCNA Security, Network Security, Cyber Security, MCSE, MCITP, Java, Spring, Hibernate, C/C++, Photoshop, Adobe Illustrator, Figma, CorelDraw etc.

 Voice: +91- 8437365007, +91-181-5015007

E-Mail: [enquiry@o7services.com](mailto:enquiry@o7services.com) , [hr@o7services.com](mailto:hr@o7services.com) Website:www.o7services.com

**INDEX**

1. **Introduction 1-2**
2. **Project Description 3-4**
3. **Analysis 5-10**
   1. Existing System
   2. Drawbacks of Existing System
   3. Proposed System
   4. Feasibility Study
   5. Requirement Analysis
4. **Hardware and Software Requirements** **11-30**
5. **Software Process Model 31-34**
6. **Design 35-40**
   1. System Design
   2. Data Flow Diagrams
   3. Database Tables
7. **Coding 41-45**
8. **Snapshots 46-48**
9. **Conclusion 49-53**
10. **Refrences 54**

**Project Title: “Veggie Delights”**

**1. INTRODUCTION:**

Project is constituent of seven words, and I have tried my best to give its definition analyzing each word in the following manner:

P- Perfect Planning

R- Resources

O- Organization

J- Joint Efforts

E- Engineering Skill

C- Communication

T- Technique

**“Veggie Delights”:**

“Veggie Delights” is an Online fruit-selling web-based application that enables customers to purchase fresh vegetables online. The project aims to make the shopping process more convenient and accessible for customers while providing a platform for hawkers and supplier to sell their produce online. This veggie delights web-application can provide significant benefits for customers and hawkers/suppliers alike.

By creating a user-friendly and feature-rich online platform, customers can access fresh produce with greater convenience and transparency, while hawkers and suppliers can expand their customer reach, increase revenue, and reduce waste. With the increase demand fresh and healthy produce, an online vegetable store projects has the potential to become a successful business model.

The web application typically has a user-friendly interface that allows customers to browse and select vegetables easily. It should have clear navigation, search functionalities, and visually appealing product displays.

An online vegetable selling web application is a platform that facilitates the buying and selling of vegetables through an internet-based interface. The web application provides a platform for vendors, such as farmers, growers, or hawker businesses, to list and sell their vegetables. Vendors can create their profiles, manage their inventory, and update product details, including prices, descriptions, and availability.

Perhaps the most important focus of the business is that it helps hawkers get better pricing when it comes to sourcing their produce. As things stand currently, the procurement process is dogged by issues. Most notably that’s a long supply chain that adds cost — prices increase as more middlemen take their cut — and means that vegetables are less fresh by the time they reach the hawker.

A web application for hawker and customer vegetable selling provides convenience to both parties. Customers can easily browse and order vegetables from the comfort of their homes, while hawkers can manage their inventory and receive orders online. The web application can offer a wide range of vegetables from various hawkers, allowing customers to explore different options and choose the ones that best suit their preferences and needs.

* User-friendly interface accessible via web browsers.
* Convenience & Wide Range of Choices.
* Access to Fresh and Quality Produce by Savings Time and Efforts.
* Availability, Inventory Management and Time-Sensitive Offers and Promotions.
* Support for Local Vendors and Hawkers.
* Customer support and growth opportunity for Hawkers and Suppliers.
* Support of Hand-to-Hand payment or Cash on delivery.
* Localization and Multilingual Support.

**2. PROJECT DESCRIPTION:**

**Modules of Project**

There are various modules associate with project. These modules are working in their specific area to lead and complete the project.

**Admin Modules of Veggie Delights:**

* **Login:**

The login is used to sign in the application. The login activity contains username and password text field and login button for login process.

* **Manage all Updates:**

In this part of module, admin add category and products record into the database. The vegetable app record includes category name, product name etc. In a vegetable selling web application, the admin plays a crucial role in managing and implementing updates to ensure the smooth functioning and improvement of the platform. Here are some key areas where the admin manages updates: Application Functionality.

* **Manage Dashboard:**

In this part of module, admin manage the dashboard of this vegetable selling web-application, like they manage user account and profiles or they can create, edit, or delete user accounts, and access user information, including contact details, order history, and preferences.

* **Manage Category and Products:**

Admins can manage the category of products and inventory of the vegetable selling application. They can add, edit, or remove products and categories, update pricing and description, and track inventory levels. The admin dashboard provides insights into product popularity, availability, and sales performance.

* **Manage list of Customers and Hawkers:**

Admins in a vegetable selling web application have access to view and manage the list of customers and hawkers. Here's how they can typically handle it:

1. **Customer List:**
   * Admin can Manage and Handle Customer Information, Search and Filter Customers, Edit Customer Profiles and Information, Handle Customer Analytics.
2. **Hawker List:**
   * Admin can Manage, Handle and View Hawker Information, Search and Filter Hawker, Edit Hawker Profiles and Vendor Verification, Managing Vendor Accounts, Vendor Performance Monitoring, Support and Assistance.

**User Module:**

* **Registration:** The very first step is Register in the app. After the successful registration user can login in the app.
* **Login:** The Second step is login. Firstly, the use will login into the site with his/her email-id and password. Implement a login mechanism that verifies the user's credentials (email and password)
* **Cart Management:** Allow users to view and edit their profile information, such as updating their name, email address, contact details, or password. Provide a dedicated profile page where users can manage their account settings.
* **Dashboard:** All the user can view their desired account data and can search the product availability according to their desired need.
* **View Category or Product:** Allow users to place vegetable orders by selecting items from their cart and providing necessary delivery details like address and contact number. Implement a checkout process that guides users through the steps of confirming the order and making payment.
* **Manage Order History:** Provide users with a section where they can view their past orders, including order details, delivery status, and invoices. This feature helps users track their purchase history and reorder items easily.
* **View Hawker’s Location:** With the help of this app, we can see the daily location of the hawkers so that the customer can go to their location and get the things they need.
* **Logout:** After the user is done with all his work, if he wants, he can come out of the app by logging out.

**Hawker’s Module:**

* **Registration:** Set up a user registration system that allows hawkers to create their own accounts. This ensures that only registered hawkers can access the admin section.
* **Login:** The Second step is login. Firstly, implement a login page where hawkers can enter their credentials (username/email and password) to authenticate themselves and gain access to the admin features.
* **Add Products to Admin:** Design a dedicated dashboard for hawkers in the admin section. This dashboard will serve as the central hub for managing their products. Within the hawker's admin dashboard, provide a form or interface where hawkers can input the details of their products. Include fields such as product name, description, category, price, quantity, and any additional attributes required for vegetables.Implement a "Add Product" button or similar action that allows hawkers to submit their product details.
* **Manage Orders:** To enable hawkers to manage orders in a vegetable selling app, including status updates such as pending, approved, declined, and delivered, you can follow these steps:
  + **Order listing:** Display a list of orders, showing important information such as order number, customer name, order date, and status. Organize the list in a clear and intuitive manner, making it easy for hawkers to identify and locate specific orders.
  + **Order status indicators:** Implement visual indicators or labels to represent different order statuses, such as pending, approved, declined, and delivered. This allows hawkers to quickly identify the current status of each order.
  + **Pending orders:** Initially, display orders with a "Pending" status in the order list. These are new orders that require the hawker's attention and decision.
  + **Reviewing orders:** When a hawker selects a pending order from the list, provide a detailed view of the order information, including the products ordered, quantities, customer details, and delivery address. Ensure all relevant information is easily accessible.
  + **Order approval:** Give hawkers the ability to approve an order if they have the requested products in stock and are ready for delivery. This action can change the order status from pending to approved.
  + **Order decline:** If a hawker cannot fulfill an order, provide an option to decline it. This action should prompt the hawker to enter a reason for declining the order. Once declined
  + **Order history:** Maintain a history of all orders processed by the hawker. This history can be accessed in the dashboard and serve as a reference for past transactions and customer interactions.
  + **Order delivered:** Once an order is delivered, update the order status as "Delivered" and mark it as completed. Maintain a record of completed orders separately for future reference.
* **Logout:** After the user is done with all his work, if he wants, he can come out of the app by logging out.

**3. ANALYSIS:**

**Problem Description:**

**3.1 EXISTING SYSTEM:**

* **Limited access to fresh vegetables:** Many people face challenges in accessing fresh vegetables, especially in urban areas where there might be a lack of nearby markets or limited availability of quality produce.
* **Inconvenience of physical markets:** Traditional physical vegetable markets often involve time-consuming trips and long queues, making it inconvenient for people with busy schedules or mobility constraints.
* **Lack of information:** Customers may not have sufficient information about the varieties, quality, and sources of vegetables they are purchasing. This can make it difficult to make informed decisions about their purchases.
* **Price fluctuations:** Vegetable prices can vary significantly depending on factors such as seasonality, demand, and supply. Customers may find it challenging to keep track of these fluctuations and find the best prices.
* **Limited vendor options:** Customers may have limited choices when it comes to vegetable vendors or may have to rely on a single vendor in their area. This restricts their options and makes it difficult to find the best quality and prices.
* **Lack of transparency:** Customers may face issues regarding the transparency of the vegetable supply chain, including information about the source, cultivation practices, and handling of the vegetables they purchase.
* **Limited access to specialty vegetables:** Customers may face challenges in finding and accessing specialty or organic vegetables that are not commonly available in physical markets.

**3.2 Proposed System:**

1. Developing a web-application dedicated to vegetable selling and buying would make it more convenient for users to buying and selling for vegetable accommodations. The app could offer features like daily hawker’s location search, user reviews, buy fresh products and ratings.
2. The proposed system could utilize machine learning algorithms to analyze user preferences and provide personalized recommendations for buying suitable products.

**3.3 Feasibility Study:**

A feasibility study evaluates a project's or system's practicality. As part of a feasibility study, the objective and rational analysis of a potential business or venture is conducted to determine its strengths and weaknesses, potential opportunities and threats, resources required to carry out, and ultimate success prospects. Two criteria should be considered when judging feasibility: the required cost and expected value.

A feasibility study is a comprehensive evaluation of a proposed project that evaluates all factors critical to its success in order to assess its likelihood of success. Business success can be defined primarily in terms of ROI, which is the amount of profits that will be generated by the project.



**Types of Feasibility:**

There are various measures of feasibility that helps to decide whether a particular project is feasible or not. These measures include-

* **Technical Feasibility Study:**

The technical issues raised during the technical feasibility analysis are:

* Does the necessary technology exist to do what is suggested?
* Does the proposal equipment have the technical capacity to hold the data required to use the new system?
* Will the proposed system & components provide adequate responses to inquiries, regardless of the number or locations of users?
* Can the system be expanded?

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn’t want to try to put Star Trek’s transporters in their building—currently, this project is not technically feasible.

* **Operational Feasibility Study:**

This assessment involves undertaking a study to analyze and determine whether—and how well—the organization’s needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

This includes the following questions:

* Is there sufficient support for the users?
* Will the proposed system cause harm?
* The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.
* **Economic Feasibility Study:**

It involves estimating cost and benefits that can be tangible and intangible because of confusing between the types of costs it is sometimes very difficult to divide the benefits out weight the cost. The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

* The costs conduct a full system investigation.
* The cost of the hardware and software.
* The costs conduct a full system investigation.
* The benefits in the form of reduced costs or fewer costly errors.
* **Legal Feasibility:**

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws,[data protection](https://www.simplilearn.com/understanding-data-security-rar30-article) acts or social media laws. Let’s say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization’s ideal location isn’t zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

* **Scheduling Feasibility:**

This assessment is the most important for [project success](https://www.simplilearn.com/how-to-make-a-project-successful-article); after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.The feasibility study examined the technology requirements of the proposed concept (new science building), the potential benefits for students, and its long-term viability. Modernizing the science facility will increase the scientific research potential and ameliorate its modules. It also would allure new students.

When these areas have all been examined, the feasibility analysis helps identify any constraints the proposed project may face, including:

* Internal Project Constraints: Technical, Technology, Budget, Resource, etc.
* Internal Corporate Constraints: Financial, Marketing, Export, etc.
* External Constraints: Logistics, Environment, Laws, and Regulations, etc.

**3.4 Requirement Analysis:**

**SYSTEM REQUIRENMENT SPECIFICATIONS:**

A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform.

An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-world situations. Parameters such as operating speed, response time, availability, portability, maintainability, footprint, security and speed of recovery from adverse events are evaluated. Methods of defining an SRS are described by the IEEE (Institute of Electrical and Electronics Engineers) specification 830-1998.

1. **Top-tier:** These are the high-level business requirements. They outline the business' measurable goals, define the purpose behind the project, and align the project goals with stakeholder goals.
2. **Middle-tier:** These are the user requirements. They reflect specific user needs and expectations, describe who is using the software, and highlight user interactions.
3. **Bottom-tier:** These specify the product's functionality in tech terms. They identify functions, features, use cases, and non-functional requirements, as well as describe the project as functional modules + non-functional attributes.



**Need for SRS:**

An SRS gives you a complete picture of your entire project. It provides a single source of truth that every team involved in development will follow. It is your plan of action and keeps all your teams — from development and testing to maintenance — on the same page.

An SRS not only keeps your teams aligned and working toward a common vision of the product, it also helps ensure that each requirement is met. It can ultimately help you make vital decisions on your product’s lifecycle, such as when to retire an obsolete feature. It takes time and careful consideration to create a proper SRS.

**Hardware and Software Requirements:**

For this project minimum hardware and software requirement are listed below:

* **Hardware Requirements:**

**Processor** **:** Intel I3

**RAM**  **:** 8.00 GB

**SSD**  **:** 256 GB

* **Software Requirements:**

**Front End Tool :** HTML, CSS, Bootstrap, JavaScript, ECMA Script, ReactJS

**DB Tool :** FireStore

**Environment :** Firebase

**Back End Tool :** Visual Studio

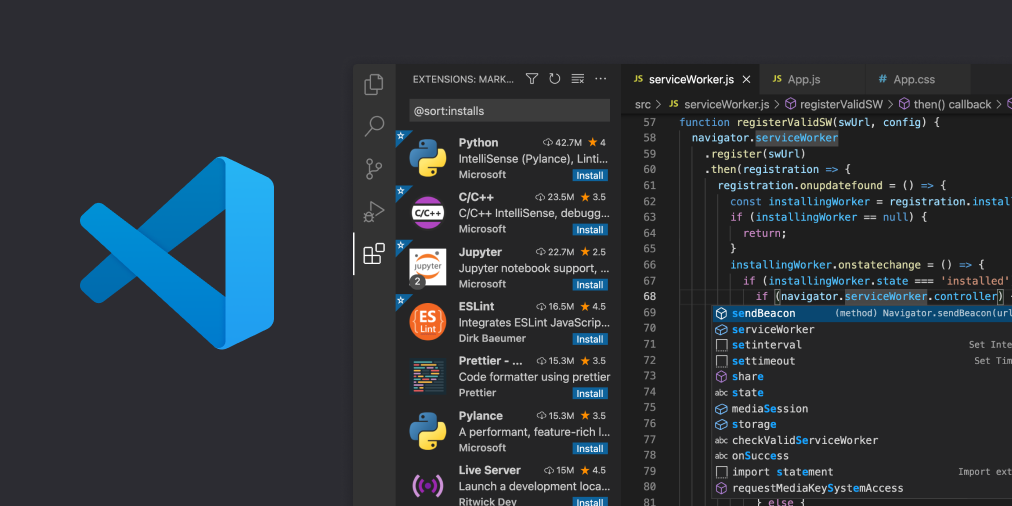
**Browser :** Mozilla Firefox/Chrome/Edge

**Operating System :** Windows Operating System/Linux

**Introduction to Firebase server:**

1. Firebase is a comprehensive mobile and web development platform provided by Google. It offers a suite of cloud-based services and tools that developers can leverage to build, deploy, and scale their applications. At its core, Firebase provides a powerful server infrastructure that enables developers to store and retrieve data, authenticate users, and host applications.
2. Firebase Server, also known as the Firebase Realtime Database, is a NoSQL cloud-hosted database service. It follows a real-time synchronization model, allowing data to be synchronized across multiple clients in real-time. This means that any changes made to the data in the database are immediately propagated to all connected clients, ensuring a consistent and up-to-date view of the data.
3. Developers can interact with the Firebase Server using client-side libraries and SDKs available for various platforms, such as JavaScript, iOS, Android, and more. These libraries provide convenient APIs to read and write data, listen for changes, and perform queries on the database. The Firebase Server supports automatic offline data persistence, allowing applications to function even when the device is offline and sync the changes when a network connection is available.
4. Overall, Firebase Server simplifies the backend development process by providing a scalable and flexible infrastructure for storing and synchronizing data in real-time. It has become a popular choice among developers for building real-time collaborative applications, chat applications, multi-user games, and other use cases that require real-time data synchronization.
5. **Firebase offers a wide range of features that empower developers to build powerful and scalable applications. Here are some key features of Firebase:**
6. Realtime Database: Firebase provides a NoSQL cloud-hosted database that enables real-time data synchronization across clients. It allows developers to build collaborative and interactive applications that instantly reflect changes made to the data.
7. Authentication: Firebase offers easy-to-use authentication services, supporting various authentication providers such as email/password, Google, Facebook, Twitter, and more. It allows developers to manage user authentication and provides secure access controls to protect data.
8. Cloud Firestore: Firestore is a flexible and scalable NoSQL document database offered by Firebase. It provides powerful querying capabilities and real-time updates, making it suitable for building complex applications that require efficient data retrieval and synchronization.

**Visual Studio Code:**



Visual Studio Code is a free coding editor that helps you start coding quickly. Use it to code in any programming language, without switching editors. Visual Studio Code has support for many languages, including Python, Java,

C++, JavaScript, and more. A standalone source code editor that runs on Windows, MacOS, and Linux. The top pick for JavaScript and web developers, with extensions to support just about any programming language.

Designed for Windows, MacOS, and Linux, Visual Studio Code offers a modern and intuitive user interface. It provides a rich set of editing capabilities, including syntax highlighting, code completion, and intelligent code suggestions that aid in writing clean and efficient code. It supports a wide range of programming languages out of the box, with built-in support for popular languages like JavaScript, Python, TypeScript, C++, and many more.

Collaboration is made easier with Visual Studio Code's Live Share extension. It allows developers to share their development environment with teammates, enabling real-time collaboration on code, debugging, and even terminal sessions.



**Properties:**

Visual Studio Code (VS Code) possesses several properties that contribute to its popularity and effectiveness as a code editor. Here are some notable properties of VS Code:

1) Lightweight and Fast.

1. Cross-Platform Compatibility.
2. Extensibility.
3. Language Support.
4. Integrated Terminal.
5. Version Control Integration.
6. Debugging Support.
7. IntelliSense.
8. Live Share.

**Advantages:**

1. **Cross-platform compatibility**: Visual Studio Code is available for Windows, macOS, and Linux, making it accessible to a wide range of developers across different operating systems.
2. **Extensibility:** VS Code has a rich ecosystem of extensions that enhance its functionality and support for different programming languages, frameworks, and tools. You can customize and extend the editor to fit your specific needs, making it highly adaptable.
3. **Lightweight and fast:** VS Code is known for its speed and efficiency. It has a small footprint and starts up quickly, enabling developers to get straight into coding without unnecessary delays. The editor's performance is optimized for smooth editing and navigation.
4. **Intelligent code editing:** Visual Studio Code provides excellent features for code editing. It offers intelligent code completion, syntax highlighting, and automatic formatting. It also supports code refactoring, debugging, and version control integration, making development tasks more productive.
5. **Integrated terminal:** VS Code includes an integrated terminal that allows you to run commands, execute scripts, and perform various command-line tasks without leaving the editor. This integrated terminal enhances workflow efficiency by eliminating the need to switch between different applications.
6. **Git integration:** Visual Studio Code has built-in Git integration, enabling seamless version control management. You can view Git changes, commit code, switch branches, and resolve conflicts within the editor. This integration simplifies collaboration and makes working with Git repositories more convenient.
7. **Powerful debugging:** VS Code provides a powerful debugging experience for various programming languages. It supports breakpoints, stepping through code, inspecting variables, and analyzing call stacks. Debugging configurations can be easily set up, allowing developers to troubleshoot and fix issues efficiently.

**Disadvantages:**

1. **Steep learning curve:** While the user interface of VS Code is generally intuitive, mastering all the features and shortcuts may require some time and effort. Especially for beginners or those coming from other code editors, there can be a learning curve to understand and utilize the full potential of the editor.
2. **Memory usage:** Although VS Code is considered lightweight compared to full-fledged IDEs, it can still consume a significant amount of memory, especially when working on larger projects or with multiple extensions. This can lead to slower performance or even lag on systems with limited resources.
3. **Limited built-in features:** VS Code is designed to be a modular and extensible editor, which means it focuses on providing a solid foundation with essential features while leaving more specialized functionalities to extensions. While this extensibility is a strength, it also means that some features.
4. **Lack of integrated build tools:** Unlike integrated development environments (IDEs) that often include build and project management tools, VS Code primarily focuses on code editing and debugging. While you can set up build tools through extensions or external command-line interfaces, the lack of built-in support might require additional configuration and setup.
5. **Plugin compatibility and stability:** The vast ecosystem of extensions available for VS Code is a significant advantage, but it can also introduce potential compatibility issues or stability concerns.
6. **Lack of comprehensive documentation:** While Visual Studio Code has documentation and a growing community, the documentation may not cover all features comprehensively or provide detailed explanations for specific scenarios. This can make it challenging to find specific information or troubleshoot certain issues.

**Features:**

1. Command Line
2. Command Palette
3. Git Integration
4. Change language mode

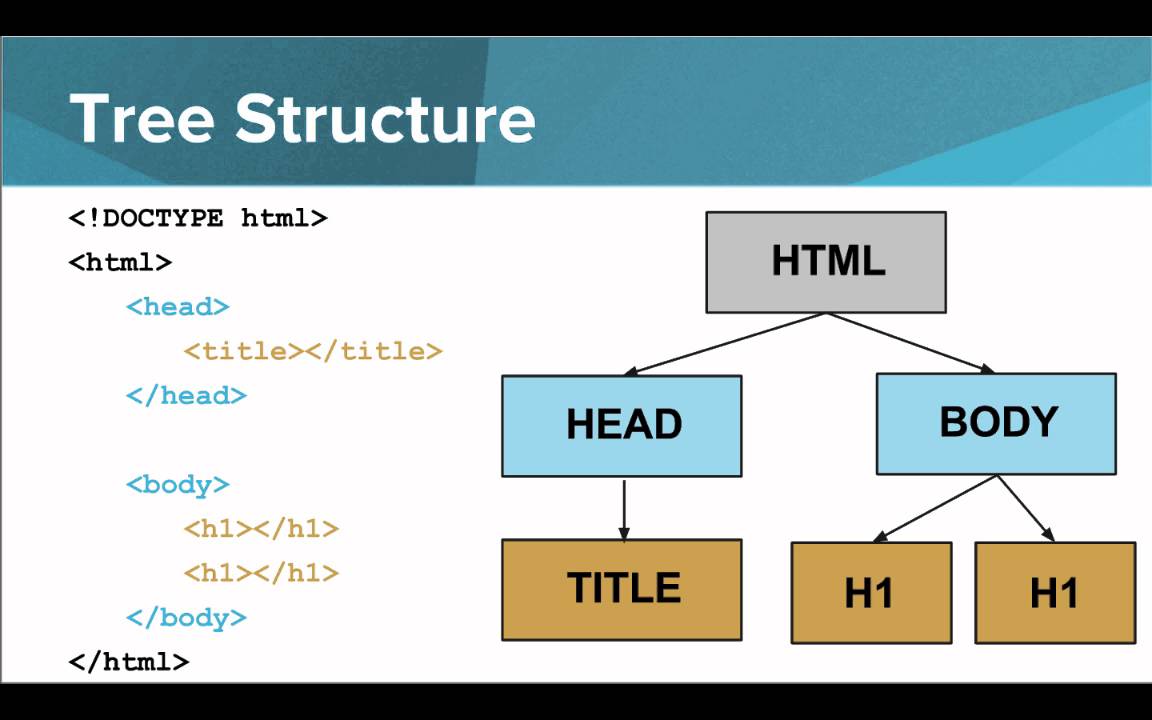
**HTML:**  
HTML stands for HyperText Markup Language. It is the standard markup language used for creating the structure and presentation of web pages on the World Wide Web. HTML uses a set of markup tags to define the elements and content within a web page.

* HTML stands for Hyper Text Markup Language
* HTML describes the structure of Web pages using markup tags
* HTML elements are the building blocks of HTML pages
* HTML elements are represented by tags
* HTML tags label pieces of content such as "heading", "paragraph", "table", and so on

**Syntax:**

* The <!DOCTYPE html> declaration at the beginning specifies the HTML version and type.
* The <html> element serves as the root element, encapsulating the entire HTML document.
* The <head> element contains metadata about the document, such as the page title specified within the <title> tags.
* The <body> element holds the visible content of the web page.

**Structure of HTML Page:**



**Example of HTML:**

<html>  
<head>  
<title>Page Title</title>  
</head>  
<body>  
<h1>My First Heading</h1>  
<p>My first paragraph.</p>

<img src=”a.jpg”/>

<a href=”next.html”>image</a>  
</body>  
</html>

**HTML Tags:**

HTML tags are element names surrounded by angle brackets:

<tag name>content goes here...</tag name>

|  |  |
| --- | --- |
| **Tag** | **Meaning** |
| <html>…..</html> | Root element of an HTML page |
| <head>…..</head> | Meta information about the document |
| <title>……</title> | Specifies the title of the document |
| <body>….</body> | Defines the document body |
| <hi>…….</h1> | Display the Headings |
| <p>……..</p> | Defines a paragraph |
| <div>…..</div> | Applying alignment and style characteristics to only a section of a document |
| <img> | Display images on the web page. It is a empty tag |
| <br> | Defines a line break |
| <meta /> | Data about data |
| <form>……</form> | Used for creating the form |
| <table>……</table> | Used for creating the tables |
| <tr>…………</tr> | Creating the row of a table |
| <td>…………</td> | Creating the columns |
| <img/> | Used for image |
| <a>…………..</a> | Used for linking the text, images etc. |
| <font>………</font> | Used for text size, color etc. |

**Properties:**

HTML (Hypertext Markup Language) is a standard markup language used for creating the structure and presentation of web pages. It provides a set of elements and attributes that define the content and layout of a webpage. Here are some key properties and characteristics of HTML:

1. **Tag-based Structure:** HTML is based on a set of tags that define different elements of a webpage. Tags are enclosed in angle brackets (< >) and can have attributes to provide additional information about the element.
2. **Document Structure:** HTML documents have a hierarchical structure consisting of an HTML root element, followed by the head and body sections.
3. **Elements:** HTML provides a wide range of elements to define different types of content, such as headings (<h1> to <h6>), paragraphs (<p>), images (<img>), links (<a>), lists (<ul>, <ol>, <li>), tables (<table>, <tr>, <td>), and many more.
4. **Cascading Style Sheets (CSS):** While HTML is primarily responsible for defining the structure of a webpage, the visual presentation is achieved using CSS. CSS allows you to control the colors, fonts, layout, and other visual aspects of HTML elements.
5. **Hyperlinks:** HTML enables the creation of hyperlinks using the anchor tag (<a>). Hyperlinks allow users to navigate between webpages, documents, or specific sections within a webpage.

**CSS:**

CSS (Cascading Style Sheets) is a stylesheet language used for describing the presentation and layout of HTML (or XML) documents. It provides a way to separate the structure and content of a webpage from its visual appearance, allowing web designers and developers to control the styling and layout of multiple web pages from a single CSS file.

**CSS Solved a Big Problem**

* HTML was NEVER intended to contain tags for formatting a web page!
* HTML was created to **describe the content** of a web page, like:
* <h1>This is a heading</h1>
* <p>This is a paragraph.</p>
* When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.
* To solve this problem, the World Wide Web Consortium (W3C) created CSS.
* CSS removed the style formatting from the HTML page!

**CSS Syntax:**

A CSS rule-set consists of a selector and a declaration block:

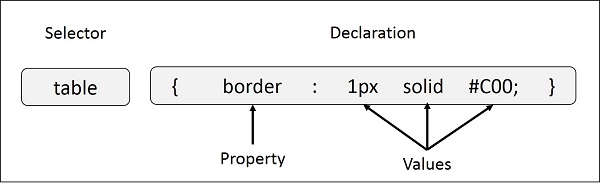


Fig 2.4.

* The selector points to the HTML element you want to style.
* The declaration block contains one or more declarations separated by semicolons.
* Each declaration includes a CSS property name and a value, separated by a colon.
* A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces

**Example:**

p {  
    color: red;  
    text-align: center;  
}

**Selectors:**

1. **The element Selector:** The element selector selects elements based on the element name.

You can select all <p> elements on a page like this (in this case, all <p> elements will be center-aligned, with a red text color):

**Example:**

p { text-align: center;  
    color: red; }

1. **The id Selector:** The id selector uses the id attribute of an HTML element to select a specific element.The id of an element should be unique within a page, so the id selector is used to select one unique element!To select an element with a specific id, write a hash (#) character, followed by the id of the element.The style rule below will be applied to the HTML element with id="para1":

**Example:**

#para1 { text-align: center;  
    color: red; }

1. **The class Selector:**

* The class selector selects elements with a specific class attribute.
* To select elements with a specific class, write a period (.) character, followed by the name of the class.
* In the example below, all HTML elements with class="center" will be red and center-aligned:

**Example:**

.center { text-align: center;  
    color: red; }

**Example:**

p.center { text-align: center;  
    color: red; }

1. **Grouping Selectors:** If you have elements with the same style definitions, like this:

h1 { text-align: center;  
    color: red; }  
h2 { text-align: center;  
    color: red; }  
p { text-align: center;  
    color: red; }

* It will be better to group the selectors, to minimize the code.
* To group selectors, separate each selector with a comma.

## RGB (Red, Green, Blue):

RGB color values can be specified using this formula: rgb(red, green, blue).

Each parameter (red, green, blue) defines the intensity of the color between 0 and 255.

# CSS Backgrounds

# The CSS background properties are used to define the background effects for elements.

# CSS background properties:

* Background-color:
* Background-image:
* Background-repeat:
* Background-attachment:

## CSS3 Linear Gradients

To create a linear gradient you must define at least two color stops. Color stops are the colors you want to render smooth transitions among. You can also set a starting point and a direction (or an angle) along with the gradient effect.

### Syntax

background: linear-gradient(direction, color-stop1, color-stop2, ...);

## Repeating a linear-gradient

The repeating-linear-gradient() function is used to repeat linear gradients:

### Example   background: -webkit-repeating-linear-gradient(red, yellow 10%, green 20%);

### CSS3 Radial Gradients

A radial gradient is defined by its center.

To create a radial gradient you must also define at least two color stops.

### Syntax

background: radial-gradient(shape size at position, start-color, ..., last-color);

By default, shape is ellipse, size is farthest-corner, and position is center.

**CSS3 Animation Properties:**

|  |  |
| --- | --- |
| **Property** | **Description** |
| @keyframes | Specifies the animation code |
| Animation | A shorthand property for setting all the animation properties |
| animation-delay | Specifies a delay for the start of an animation |
| animation-direction | Specifies whether an animation should play in reverse direction or alternate cycles |
| animation-duration | Specifies how many seconds or milliseconds an animation takes to complete one cycle |
| animation-fill-mode | Specifies a style for the element when the animation is not playing (when it is finished, or when it has a delay) |
| animation-iteration-count | Specifies the number of times an animation should be played |
| animation-name | Specifies the name of the @keyframes animation |
| animation-play-state | Specifies whether the animation is running or paused |
| animation-timing-function | Specifies the speed curve of the animation |

* **Three Ways to Insert CSS**
* External style sheet
* Internal style sheet
* Inline style
* **External Style Sheet**

With an external style sheet, you can change the look of an entire website by changing just one file!

Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section:

**Example**

<head>  
<link rel="stylesheet" type="text/css" href="mystyle.css">  
</head>

* **Internal Style Sheet**

An internal style sheet may be used if one single page has a unique style.

Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

**Example**

<head> <style> body { background-color: linen; }  
h1 { color: maroon;

margin-left: 40px; }   
</style> </head>

* **Inline Styles**

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

The example below shows how to change the color and the left margin of a <h1> element:

**Example**

<h1 style="color:blue;margin-left:30px;">This is a heading</h1>

**Advantage:**  
1. Time Saving  
2. make our website attraction looking  
3. quickly loading the pages

**Disadvantage:**

* it will be changing the alignment with different browser .
* Browser compatibility(some style sheet are supported and some not)

**JavaScript:**

JavaScript is a high-level programming language primarily used for adding interactivity and dynamic behavior to web pages. It is supported by all modern web browsers and allows developers to create interactive features, perform calculations, manipulate HTML elements, handle events, and communicate with servers.

## JavaScript Can Change HTML Styles (CSS)

Changing the style of an HTML element, is a variant of changing an HTML attribute:

* A **computer program** is a list of "instructions" to be "executed" by the computer.
* In a programming language, these program instructions are called **statements**.
* JavaScript is a **programming language.**
* **Features of JavaScript**

JavaScript is a client side technology, it is mainly used for gives client side validation, but it have lot of features which are given below:-

* JavaScript is a object-based scripting language.
* Giving the user more control over the browser.
* It Handling dates and time.
* It detecting the user's browser and OS.
* It is light weighted.
* JavaScript is a scripting language and it is not java.
* JavaScript is interpreter based scripting language.
* JavaScript is case sensitive.

### Advantages of JavaScript:

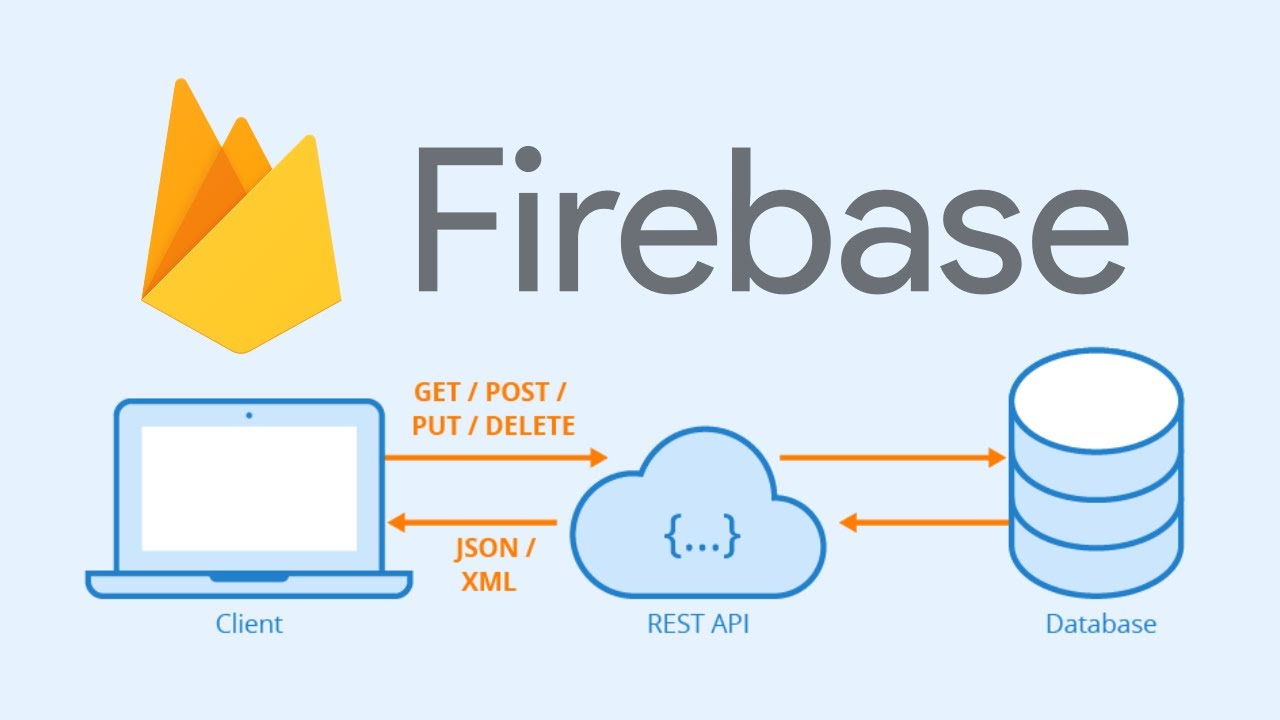
* **Speed.** Being client-side, JavaScript is very fast because any code functions can be run immediately instead of having to contact the server and wait for an answer.
* **Simplicity.** JavaScript is relatively simple to learn and implement.
* **Versatility.** JavaScript plays nicely with other languages and can be used in a huge variety of applications. Unlike PHP or SSI scripts, JavaScript can be inserted into any web page regardless of the file extension. JavaScript can also be used inside scripts written in other languages such as Perl and PHP.
* **Server Load.** Being client-side reduces the demand on the website server.
* **JavaScript is executed on the client side.** This means that the code is executed on the user's processor instead of the web server thus saving bandwidth and strain on the web server.
* **JavaScript is a relatively easy language.** The JavaScript language is relatively easy to learn and comprises of syntax that is close to English. It uses the DOM model that provides plenty of prewritten functionality to the various objects on pages making it a breeze to develop a script to solve a custom purpose.

### Disadvantages of JavaScript:

* **Security.** Because the code executes on the users' computer, in some cases it can be exploited for malicious purposes. This is one reason some people choose to disable JavaScript.
* **Reliance on End User.** JavaScript is sometimes interpreted differently by different browsers. Whereas server-side scripts will always produce the same output, client-side scripts can be a little unpredictable. Don't be overly concerned by this though - as long as you test your script in all the major browsers you should be safe.
* **Security Issues.** JavaScript snippets, once appended onto web pages execute on client servers immediately and therefore can also be used to exploit the user's system. While a certain restriction is set by modern web standards on browsers, malicious code can still be executed complying with the restrictions set.

**BACK END:**

**Firebase:**

****

Firebase is a comprehensive platform developed by Google that provides various services and tools for building and managing web and mobile applications. Here are 10 key points about Firebase:

Real-time Database: Firebase offers a NoSQL, cloud-hosted database that allows real-time synchronization of data across multiple clients. It enables developers to build applications with real-time collaboration, chat functionality, live updates, and offline capabilities.

Authentication: Firebase provides user authentication and identity management services, supporting popular authentication methods like email and password, social logins (Google, Facebook, etc.), and third-party authentication providers. It simplifies the process of integrating user authentication into your application.

Cloud Storage: Firebase offers cloud storage for storing and serving user-generated content such as images, videos, and other files. It provides an easy-to-use API for uploading, downloading, and managing files in the cloud, reducing the complexity of managing file storage infrastructure.

Cloud Functions: Firebase allows you to deploy serverless functions in the cloud, known as Cloud Functions for Firebase. These functions respond to events from Firebase services or HTTP requests and can be used for tasks such as data processing, integration with third-party services, and triggering actions based on specific events.

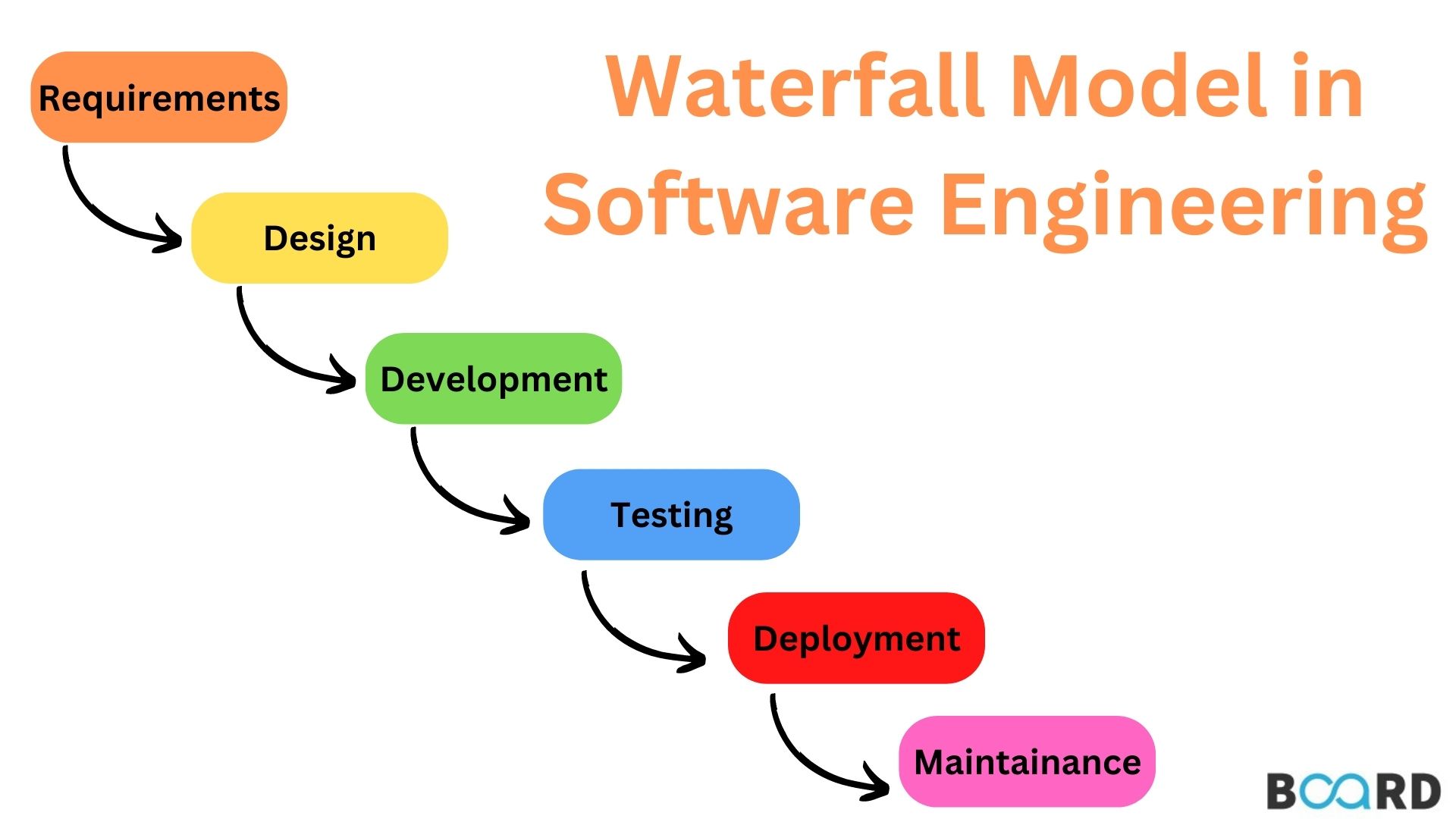
Analytics: Firebase Analytics helps you understand user behavior and app usage. It provides insights into user engagement, conversion rates, retention, and other key metrics, enabling you to make data-driven decisions to improve your application.

**5.SOFTWARE PROCESS MODEL:**

**Waterfall Model**

The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Requirement Analysis, Design, Coding, Testing, Deployment, and Maintenance .

Following is a diagrammatic representation of different phases of waterfall model.



The sequential phases in Waterfall model are:

* **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase. Requirements are set of functionalities and constraints that the end-user (who will be using the system) expects from the system. The requirements are gathered from the end-user by consultation, these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied.
* **System Design:** Before a starting for actual coding, it is highly important to understand what we are going to create and what it should look like? The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
* **Implementation:** With inputs from system design, the work is divided in modules/units and actual coding is started. The system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing. Unit testing mainly.
* **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. These units are integrated into a complete system during Integration phase and tested to check if all modules/units coordinate between each other and the system as a whole behaves as per the specifications. Post integration the entire system is tested for any faults and failures.
* **Maintenance:** This phase of "The Waterfall Model" is virtually never ending phase. There are some issues which come up in the client environment. Not all the problems come in picture directly but they arise time to time and needs to be solved. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

**Waterfall Model Application:**

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

* Requirements are very well documented, clear and fixed.
* Product definition is stable.
* Technology is understood and is not dynamic.
* There are no ambiguous requirements.
* Ample resources with required expertise are available to support the product.

## Advantage:

The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

### Disadvantage:

### The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage. Not suitable for the projects where requirements are at a moderate to high risk of changing.

**Why we use waterfall model?**

As it is a major project and being a beginner, we already have the requirements for our ongoing project. Waterfall model is considered to be of downward approach and we don’t have to look up to the previous level that frequently, it’s beneficial for our project to complete it in a timely manner.Thus if we want to modify anything within our project after deployment, we can start from the initial phase. Thus it does not freeze the possibility for any kind of change.

**6. DESIGN**

**6.1 SYSTEM DESIGN**

The most creative and challenging phase of SDLC is system design. The term design describes a final system and the process by which it is developed. It includes construction of programs and program testing.The purpose of the design phase is to plan a solution of the problem specifies by the requirements document. This phase is the first step in the moving from the problem domain to the solution domain. Starting with what is needed; design takes us towards how to satisfy the needs. The design of the system is perhaps the most critical factor affecting the quality of the software. It has major impact on the later phase, particularly testing and maintenance. The output of this phase is the design document. This document is similar to blueprint or plan for the solution and is used later during implementation, testing and maintenance.

A systematic method has to achieve the beneficial result at the end. It includes starting with average idea and developing it into a series of steps. The series of steps for successful system development are given below:

* Study problem completely because first of all we should know the goal, which he has to achieve.
* We should see what kind of output we require and what kind of input we give so we can get the desired output from system output from system. It is very challenging step of system development.
* According to the output requirement of system the strength of various databases should be design.
* Next, we should know what kind of program we should develop, which will lead us to reach final goal.
* Then we write this individual program, which later on joining solve problem.
* Then we test these programs and make necessary correction in them to achieve target of program.
* At last combining all these problems in the forms of a bar in the menu of windows, this will complete software package for general insurance.

The three main objectives which the designer has to bear in mind are:-

1. How fast the design will be do the users work given particular hardware resources.
2. The extent to which the design is secure against the human errors and machine malfunctions.
3. The ease with which the design allows the system to be changed.

**To meet these objectives analyst and programmers use a top-down and bottom-up design.**

* **TOP – DOWN DESIGN**

It is also known as system design, aims to identify the modules that should be in a system. It starts with large picture and move to the details. The analyst and team members look at major functions that the system must provide and break these down into smaller and smaller activities.

* **BOTTOM – UP APPROACH**

It is also known as detailed design. It starts with details and then moves to the big picture. This approach is appropriate when users have specific requirements for output.

**6.2 DFD: Data Flow Diagram**

Data Flow Diagrams were first developed by Larry Constantine as a way of expressing system requirements in a graphical form. DFD is also known as bubble chart and has a purpose of clarifying system requirements and identifying major transformations and will become the program in the system design.

**Purpose:**

The purpose of data flow diagrams is to provide a semantic bridge between users and systems developers.

The diagrams are:

* Graphical, eliminating thousands of words.
* Logical representations, modeling what a system does, rather than physical models showing how it does it.
* hierarchical, showing systems at any level of detail and

**DFD Symbols are as follows:**

* The External Entity symbol represents sources of data
* to the system or destinations of data from the system.
* The Data Flow symbol represents movement of data.

* The Data Store symbol represents data that is not moving (delayed data at rest).
* The Process symbol represents an activity that transforms or manipulates the data.

**DFD:**

**Context Level Diagram:**

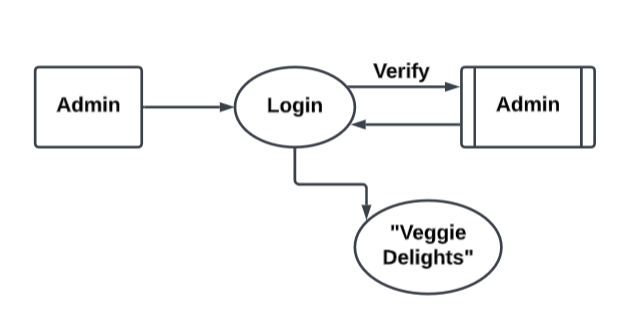
**USER**

**ADMIN**

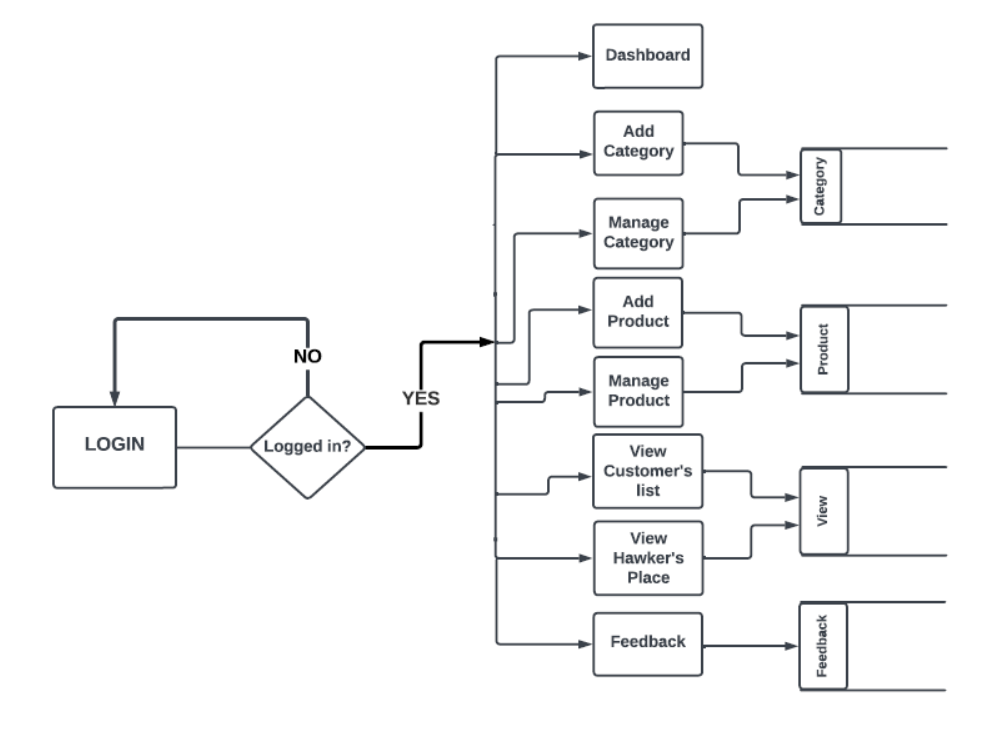
**HAWKER**

Here User and Admin interact with the system for different purpose. Database contains all information which user need.

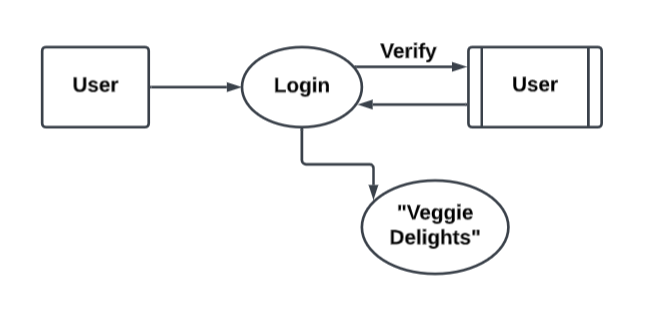
**Level 0 DFD for Admin:**

****

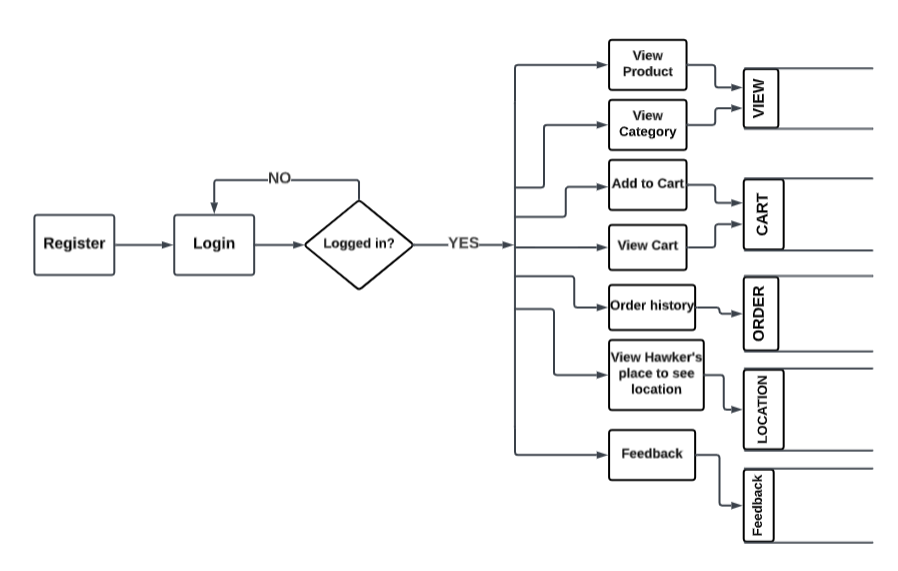
**Level 1 DFD for Admin :**



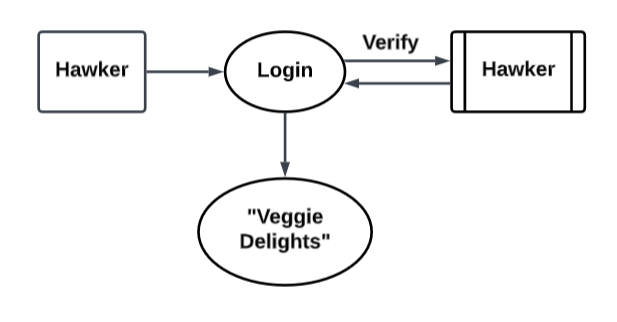
**Level 0 DFD for User :**



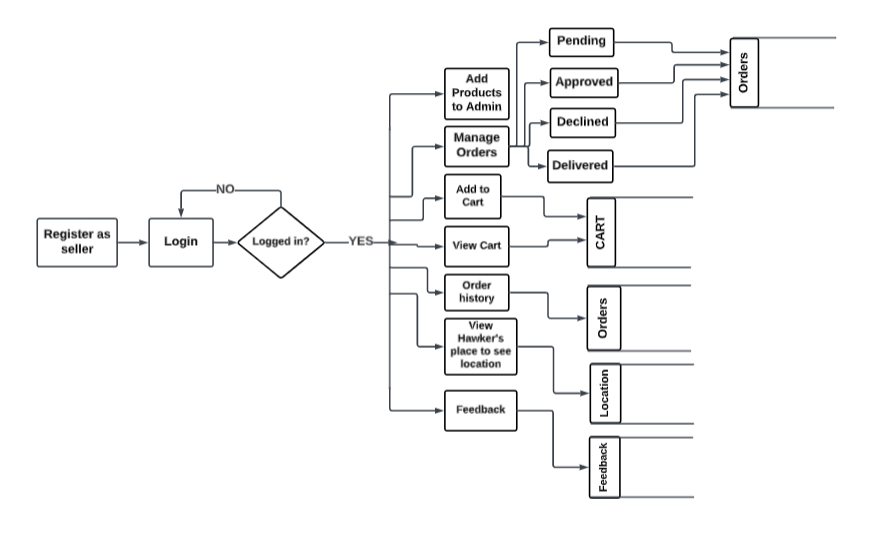
**Level 1 DFD for User :**

****

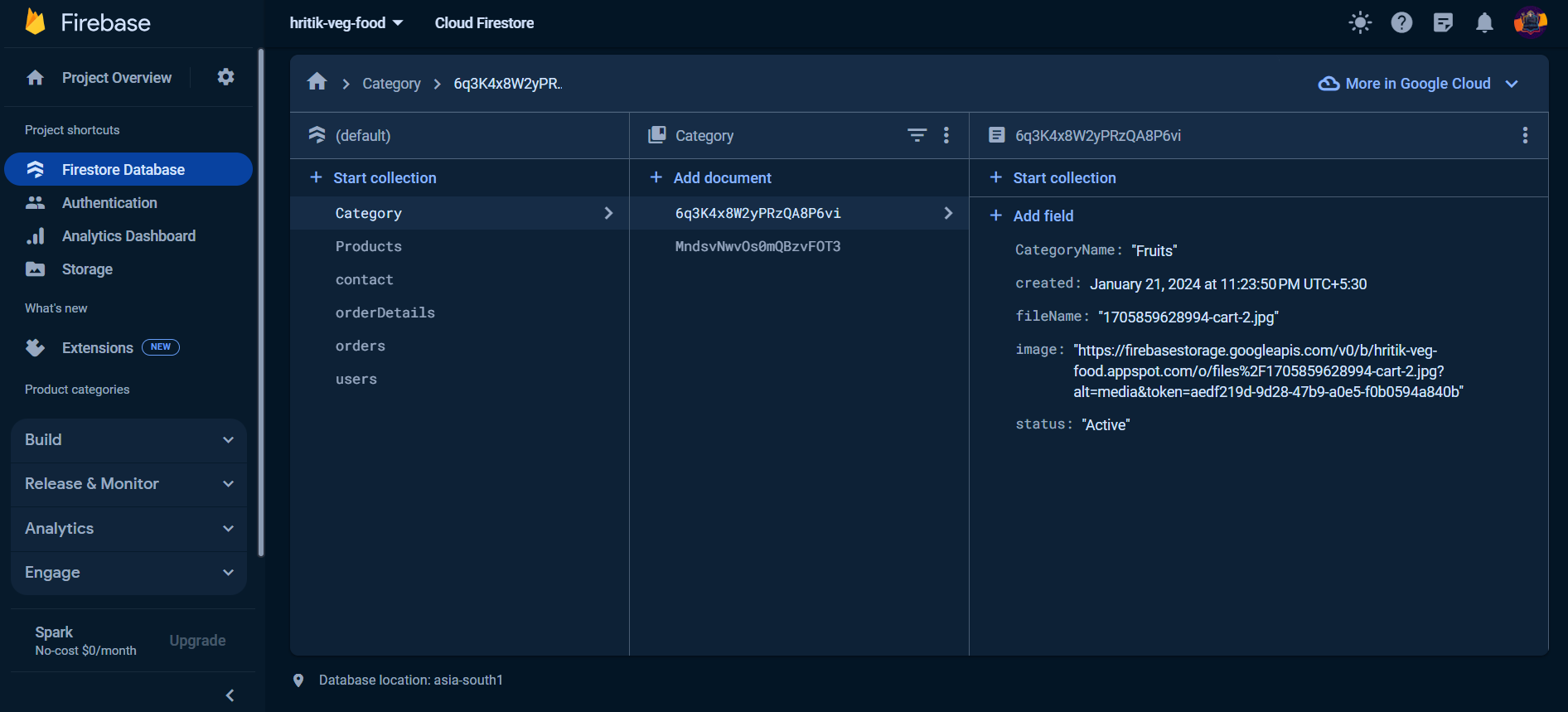
**Level 0 DFD for Hawker :**

****

**Level 1 DFD for Hawker :**

****

**6.4Database Tables:**





**7*.* CODING:**

**Header Page**

<!--

Author: W3layouts

Author URL: http://w3layouts.com

License: Creative Commons Attribution 3.0 Unported

License URL: http://creativecommons.org/licenses/by/3.0/

-->

<!DOCTYPE html>

<html lang="en">

<head>

<title>Official Corporate Category Flat Bootstrap Responsive Website Template | Home : W3layouts</title>

<!-- Meta tag Keywords -->

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta charset="utf-8">

<meta name="keywords" content="Official Responsive web template, Bootstrap Web Templates, Flat Web Templates, Android Compatible web template,

Smartphone Compatible web template, free webdesigns for Nokia, Samsung, LG, SonyEricsson, Motorola web design" />

<script>

addEventListener("load", function () {

setTimeout(hideURLbar, 0);

}, false);

functionhideURLbar() {

window.scrollTo(0, 1);

}

</script>

<!--// Meta tag Keywords -->

<link rel="stylesheet" href="css/bootstrap.css"><!-- Bootstrap-Core-CSS -->

<link rel="stylesheet" href="css/style.css" type="text/css" media="all" /><!-- Style-CSS -->

<link rel="stylesheet" href="css/font-awesome.min.css" type="text/css" media="all" />

<!-- Style-CSS --> <!-- //css files --> <!--web font--> <link <!-- //header -->

<!-- banner -->

<div class="banner\_w3lspvt" id="home"> <div class="container">

<div class="row banner-tops-style"> <div class="col-lg-8 style-banner">

<h3 class="text-wh"> Creative Agencies <br> Also tends to Cover</h3>

<p class="text-li mt-4">Praesent at molestienibh, eulaoreetmassa. Nam tristiquetortorrisus, vitae ornarediamtincidunt vitae. In in porta arcu. Integer non convallis mauris. Quisque at nunc at nibhdapibusrutrumveleumetus. Nullamlaoreetvulputatetortor.</p>

<a href="about.php" class="btn button-style mt-sm-5 mt-4">Know About Us</a>

<a href="#" class="btn call mt-sm-5 mt-4"><span class="fa fa-phone" aria-hidden="true"></span> Call: +012 898 909 2317</a> </div> </div> </div> </div>

<!-- //banner -->

**Footer Page:**

<!-- brands -->

<section class="brands py-5" id="brands">

<div class="container py-lg-0">

<div class="row text-center">

<div class="col-sm-2 col-3">

<a href="#"><span class="fa fa-connectdevelop" aria-hidden="true"></span></a> </div>

<div class="col-sm-2 col-3">

<a href="#"><span class="fa fa-first-order" aria-hidden="true"></span> </a> </div>

<div class="col-sm-2 col-3 mt-sm-0 mt-4">

<a href="#"> <span class="fa fa-joomla" aria-hidden="true"> </span> </a> </div>

<div class="col-sm-2 col-3 mt-sm-0 mt-4">

<a href="#"> <span class="fa fa-dropbox" aria-hidden="true"> </span> </a> </div></div>

</div> </section> <!-- brands --> <!-- footer -->

<footer class="py-5">

<div class="container py-md-3">

<div class="row footer-grids">

<div class="col-md-4">

<div class="footer-grid left">

<h2 class="logo"><a href="index.php"><span class="fa fa-folder-open-o"></span>Official</a></h2> </div> </div>

<div class="col-md-4 middle">

<p class="btn call"><span class="fa fa-phone"></span>Call: +012 898 909 2317</p> </div>

<div class="col-md-4 right"> <!-- to top -->

<div class="top-icon">

<a href="#home" class="move-top text-center"><span class="fa fa-angle-up mb-3" aria-hidden="true"> </span>To Top </a> </div> <!-- //to top -->

**Login Page:**

<?php include 'header1.php';

if(isset($\_SESSION["studentsession"]))

echo "<script>window.location.assign('welcome.php')</script>";

<div class="container">

<div class="row">

<div class="col-md-4"></div>

<div class="col-md-4">

<center><h1>Student Login</h1></center> <?php

if(isset($\_REQUEST["msg"]))

<div class="alert alert-info alert-dismissible fade show" role="alert">

<strong><?php echo $\_REQUEST["msg"];?></strong>

**8. SNAPSHOTS :**

Fig 8.1 Admin Login

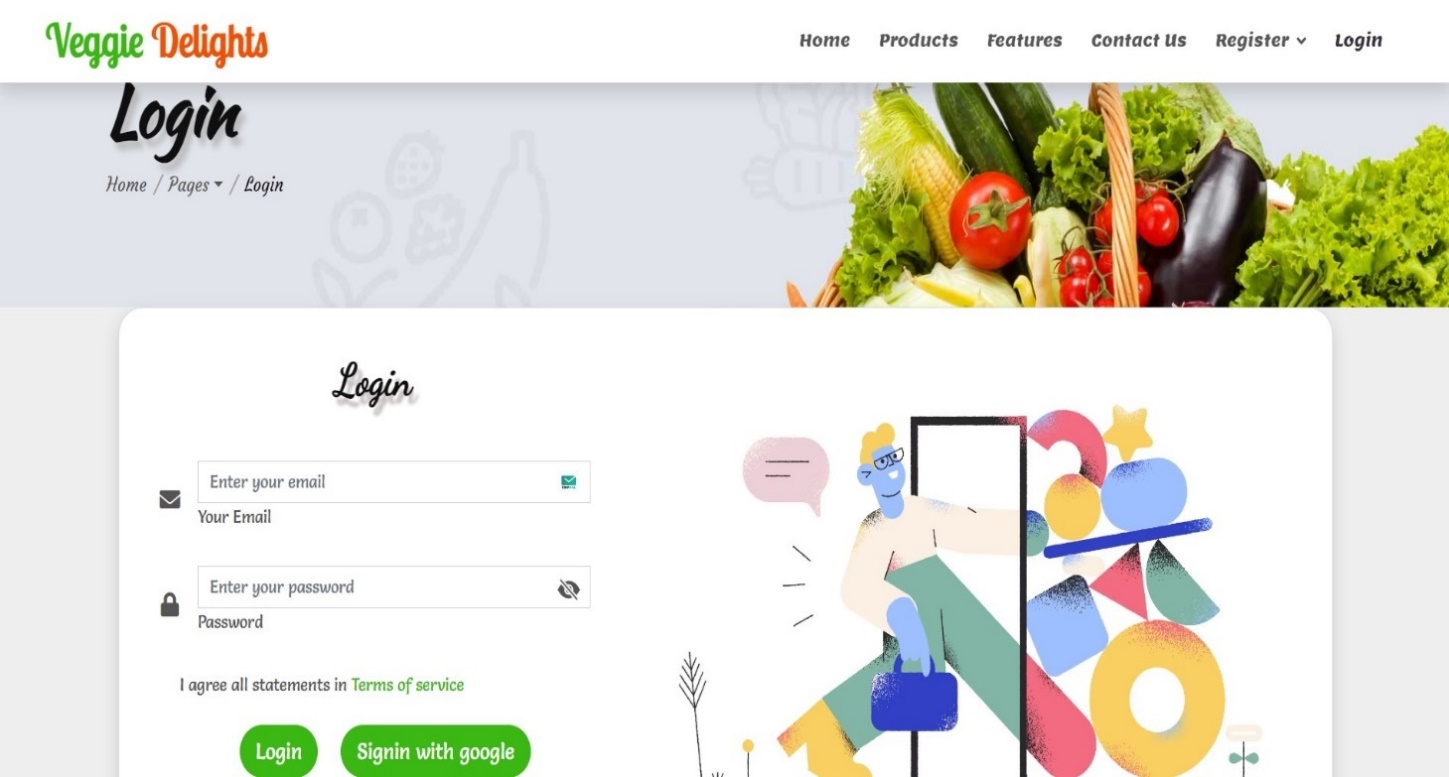


Fig 8.2 Admin Dashboard

A screenshot of a food menu

Description automatically generated

Fig 8.3 Add Products

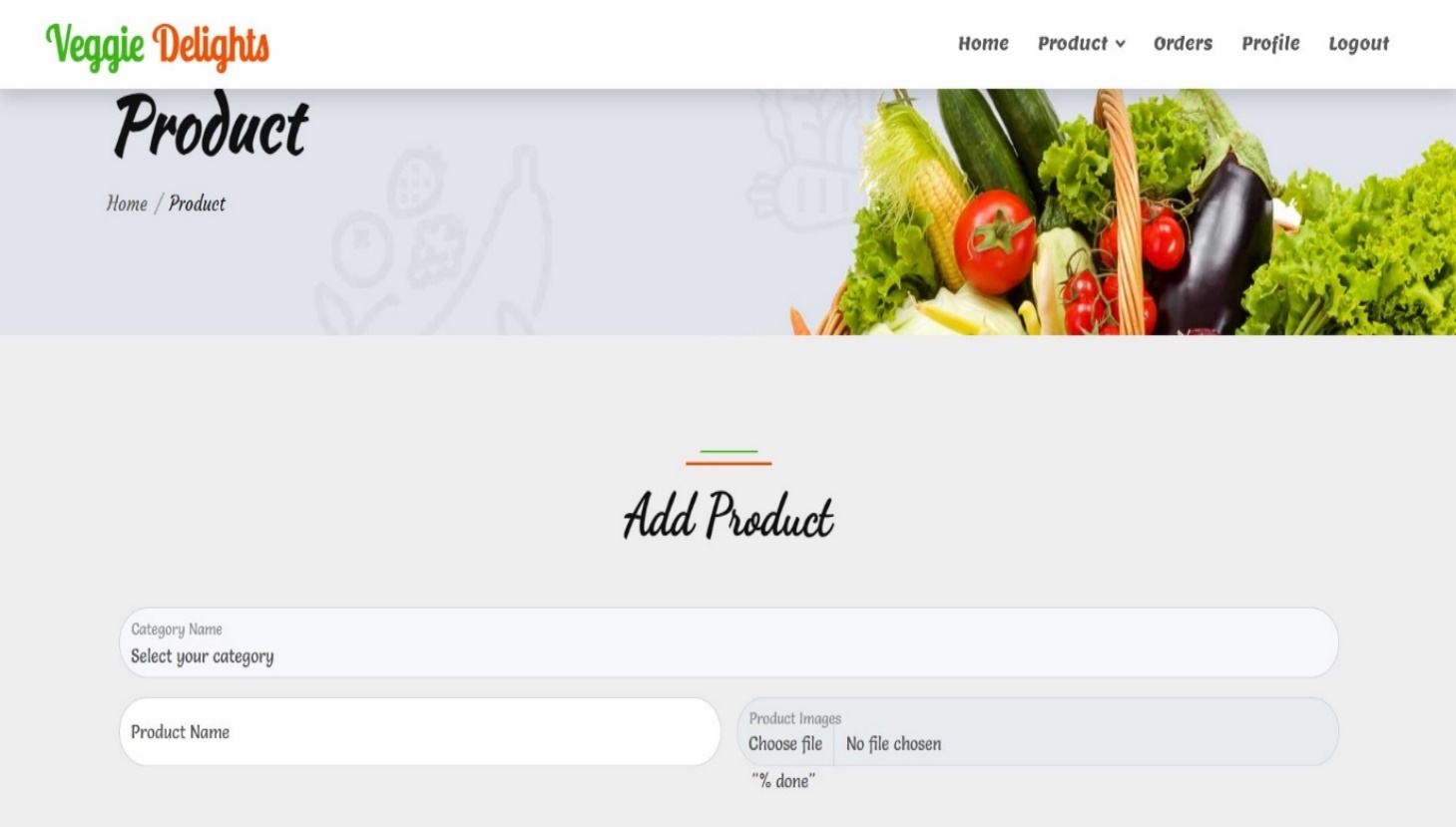


Fig 8.4 Add Category

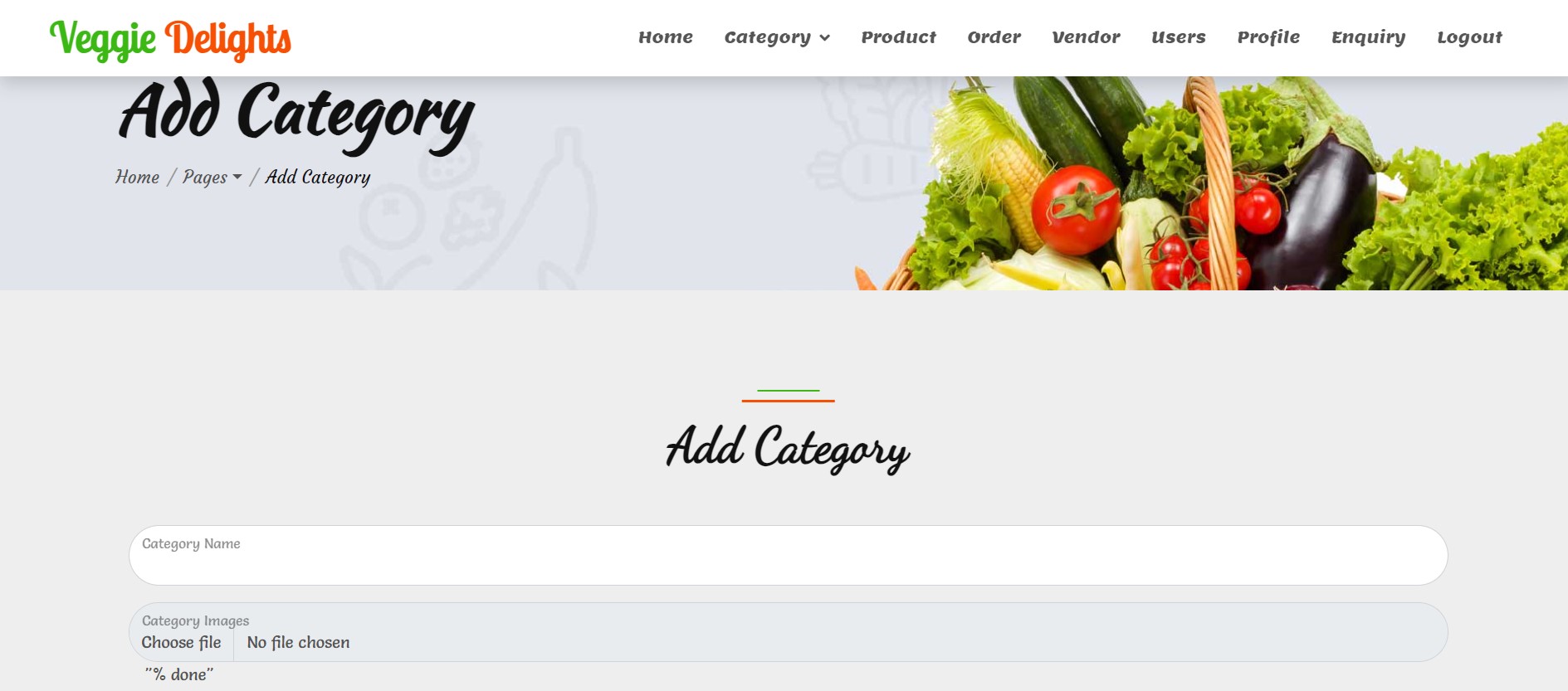


Fig 8.5 Manage Products

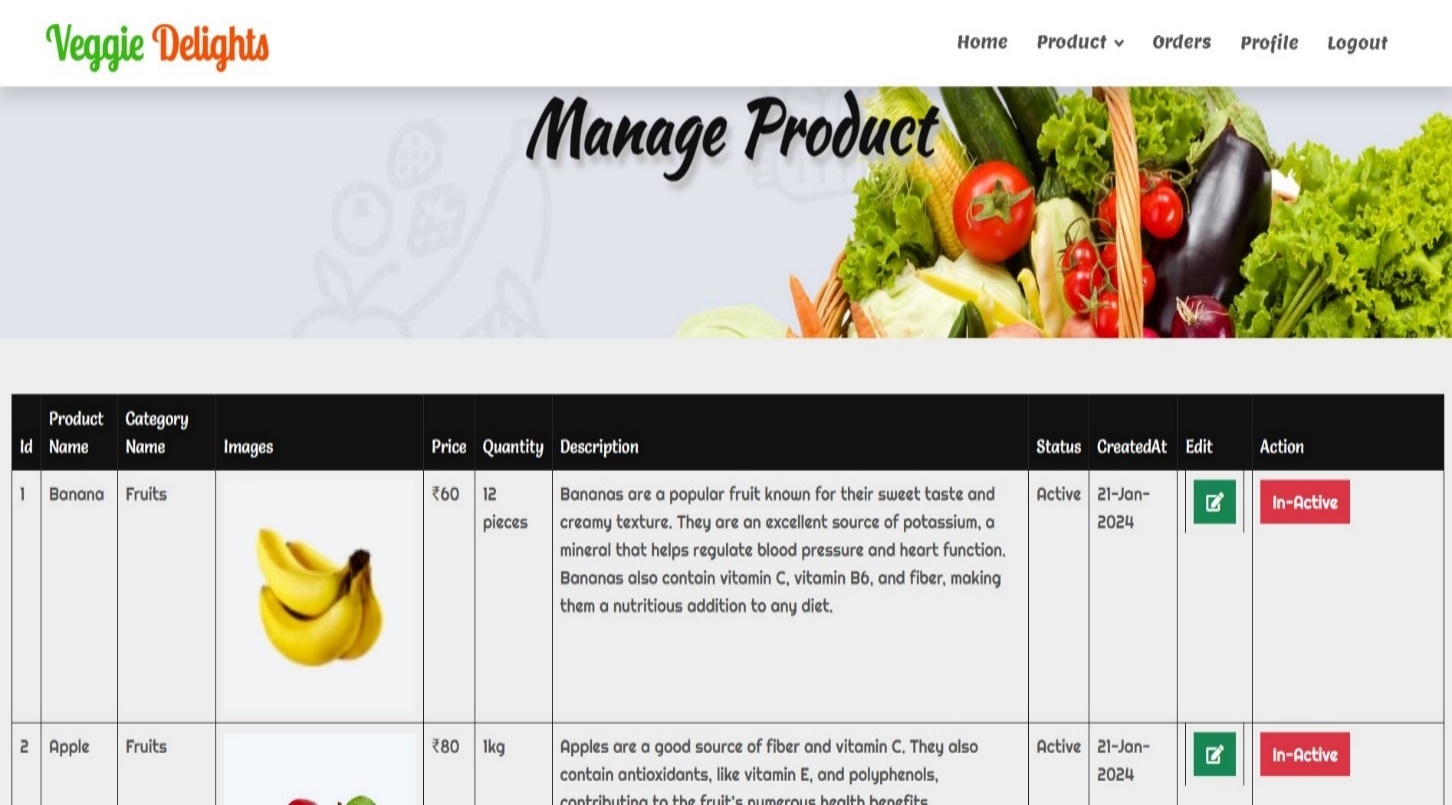


Fig 8.6 Manage Category



Fig 8.7 Add to Cart

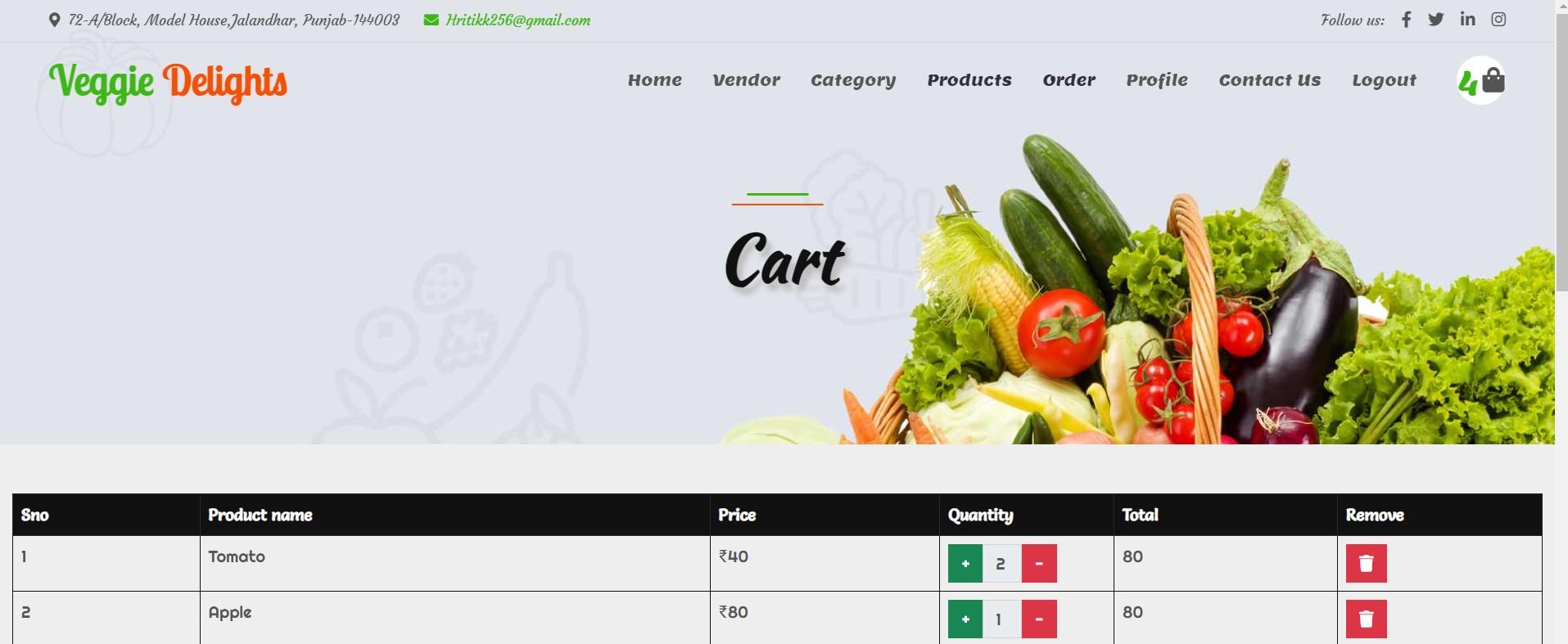


Fig 8.8 Manage Cart

A screenshot of a computer

Description automatically generated

Fig 8.9 View Hawker’s list

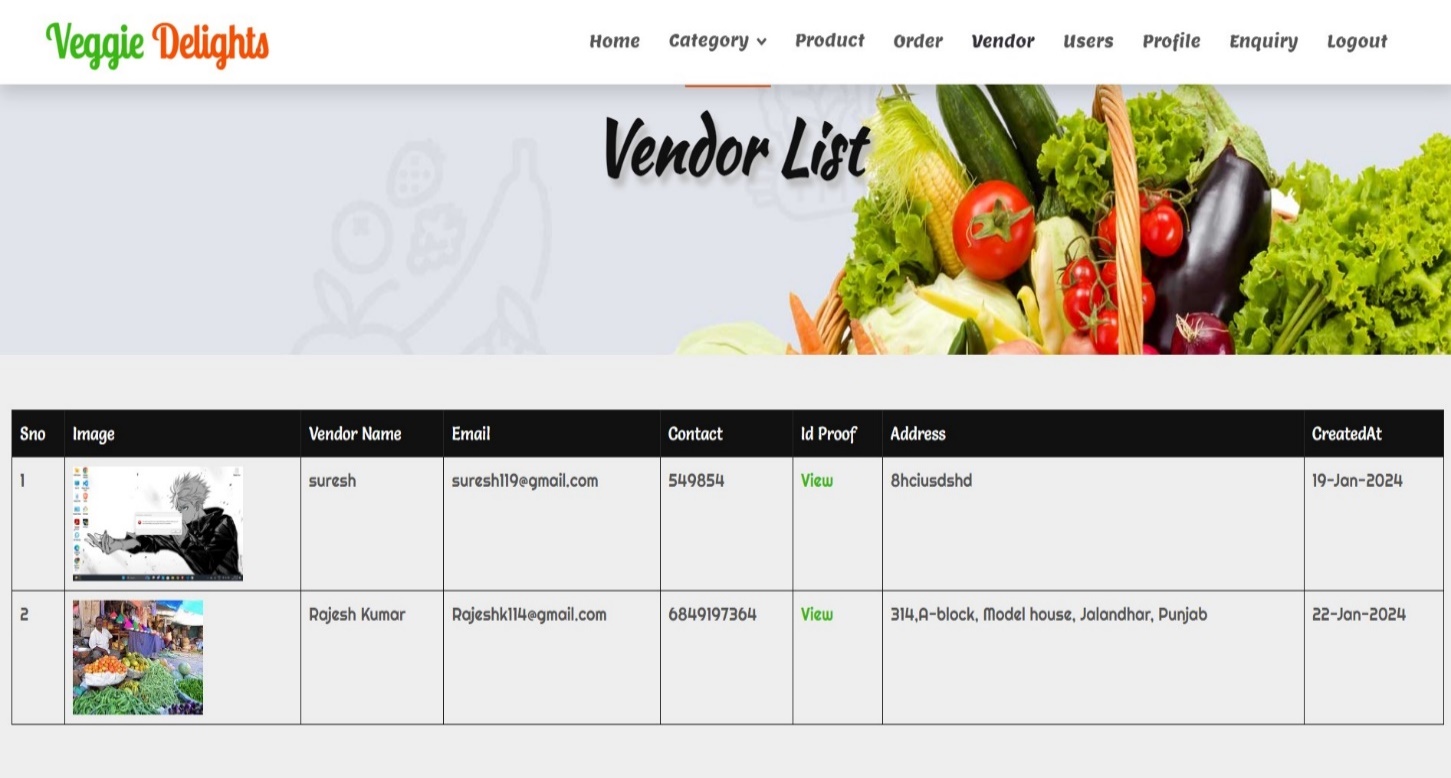


Fig 8.10 View Customer’s list

A screenshot of a website

Description automatically generated

Fig 8.11 Place Order

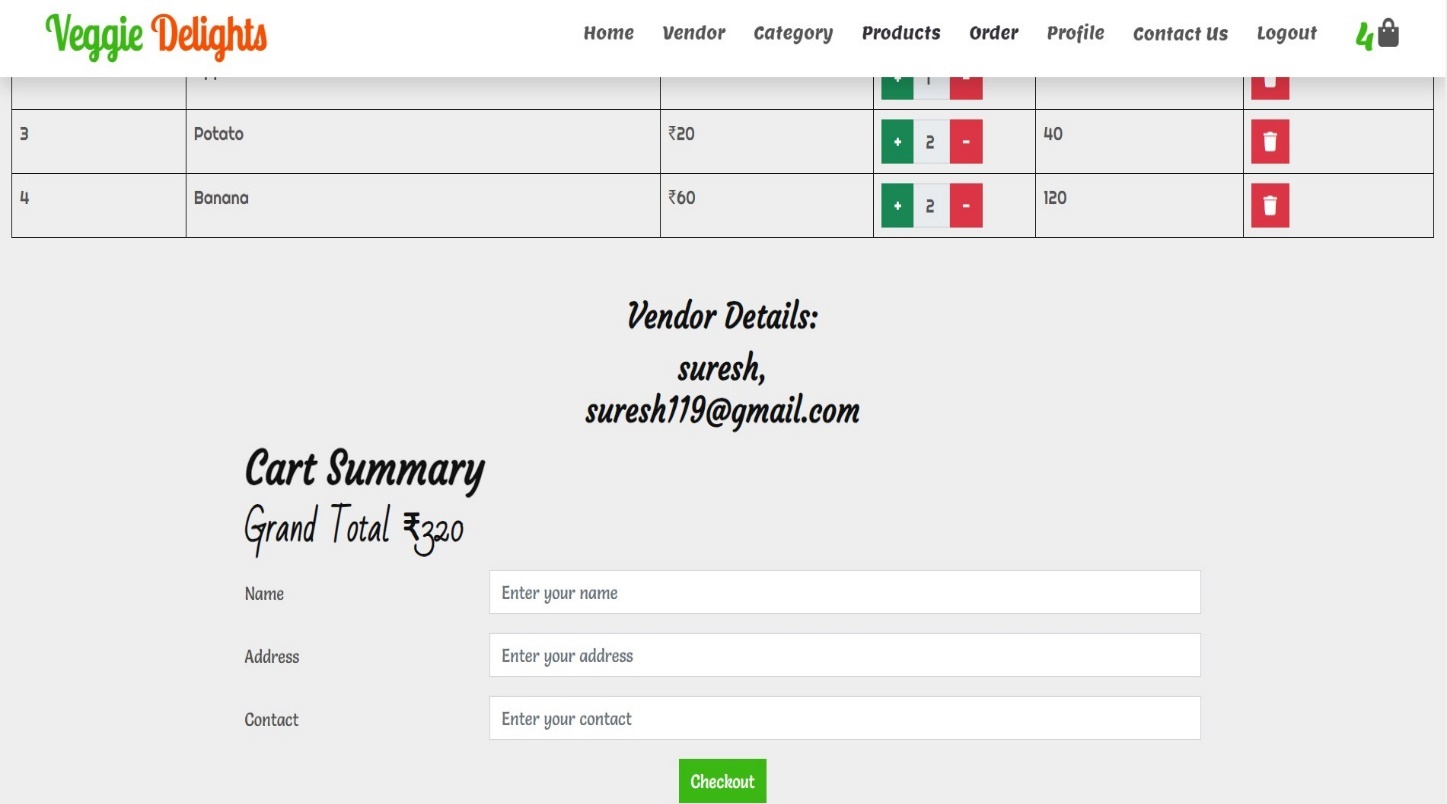


Fig 8.12 Check Orders

A screenshot of a website

Description automatically generated

Fig 8.13 View Pending Orders

A screenshot of a website

Description automatically generated

Fig 8.14 View Hawker’s location

A map of the united states

Description automatically generated

Fig 8.15 Order History

A screenshot of a website

Description automatically generated

Fig 8.16 Logout

A screenshot of a phone

Description automatically generated

**9.CONCLUSION:**

**9.1 Conclusion**

The development of the “Veggie Delights” using React has been a significant endeavor in creating a comprehensive platform for accessing and exploring information about various topics & problems of normal peoples. The project successfully achieved its objective of providing a userfriendly and interactive interface for users to access a vast repository of products.

Throughout the development process, careful consideration was given to the design, implementation, and integration of the frontend and backend components. The utilization of React, along with JavaScript & Firebase, provided a solid foundation for building a scalable and efficient application.

**9.2 Maintenance**

Maintenance is a crucial aspect of any software project, and the “Veggie Delights” is no exception. Regular maintenance activities, such as bug fixing, performance optimization, and security updates, are essential to ensure the app's smooth operation and user satisfaction. It is recommended to establish a maintenance plan and allocate resources to address any potential issues and continuously improve the app's performance and user experience.

**10. REFRENCES:**

**Websites:**

* [www.google.com](http://www.google.com)
* [www. https://themewagon.com](http://www.guruman.com)
* [www. https://console.firebase.google.com](www.%20https://console.firebase.google.com)
* [www. https://fonts.google.com](http://www.google.com)
* [www. https://www.blackbox.ai](http://www.google.com)
* [www. https://stackoverflow.com](http://www.google.com)
* [www. https://elements.envato.com/web-templates](http://www.google.com)
* [www. https://www.canva.com/website-builder/templates](http://www.google.com)
* [www. https://themeforest.net/category/all](http://www.google.com)