Software Requirements Specification

for

HUNGRY

a *Food Ordering System*

prepared by **Jagannathan Mahalingam, Mughil Srinivasan R S, Kesavan T, Kishore V.**

Table of Contents

1. Introduction

1.1 Purpose

This SRS document presents a detailed description of the Food Ordering system, version 1.0. It represents the client requirements analysis that defines the functional and non-functional requirements of the food-ordering website and its different functionalities. It defines the abilities, reactions from stimuli, guidelines, and limitations of the system. This document will be complete in its scope of the system and the functions required. The system provides a solution for users to search for the available food, order them from either the single dealer or multiple dealers at the same time, and provide feedback. It also helps the dealers to enter the available dishes, edit their menu cards, accept orders, and view feedback. It has been developed in such a way that it suits the college community.

1.2 Document Conventions

The document follows the IEEE format standard (IEEE Std. 830 – 1998).

1.3 Intended Audience and Reading Suggestions

The intended audiences of this document are Dr. Oswald C, who is the Lecturer of, the Software Engineering Course, CSPE41 Software Engineering Class, and anyone who has an interest in Software Engineering.

1.4 Project Scope

This Food Delivery System Software allows both the users and the dealers to use it. Dealers will have the option to upload their menu-card and when a user places an order to them they can view their orders and the feedback that they received from them. From the user perspective, it gives an option to order food from the available dealers and they can submit the feedback to their dealer. Dealer can add/remove/update the items from their menu card based on availability. They can also cancel an order in case of some unforeseen circumstances of not being able to deliver the food. Users on the other hand can add, and remove items from the carts, order from multiple dealers at a time, and submit feedback.

1.5 References

- Pressman, Roger S. Software Engineering: A Practitioner's Approach. New York, NY: McGraw-Hill, 2005.
- Lecture slides

2. Overall Description

2.1 Product Perspective

This project represents the initial version of the HUNGRY food delivery system. All requirements listed herein describe a self-contained system. At a high level, this project will allow a user to view, and order food from a variety of retailers as well as update orders and provide feedback. The project also allows retailers to add, update, and delete items in their menu, view feedback, orders, and users, and delete orders. The goal is to allow customers greater and easier access to high-quality food provided by people willing to cook fresh and at home.

2.2 Product Features

We can subdivide the project into 7 main features. Details of each of the following functions can be found in Section 3.

2.2.1 System Features

2.2.1.1 Login

Description: This function allows a registered user to log in to his account using a username and password. If a user is not registered, the website shall allow the user to sign up first. The system will check both the username and password when a user attempts to log in.

Rationale: This provides security to the system by authenticating each member and provides confidence to the consumer that his/her personal information is secure.

2.2.1.2 Register

Description: This function allows unregistered users to sign up and create a new account. To create a new account, the user must provide the required information such as username and password.

Rationale: A user who wishes to order and sell food he or she must be logged in. However, without registering, a user can never be a member. This section offers all users a chance to become a member.

2.2.2 Seller Features

2.2.2.1 Viewing Menu Card

Description: This function allows sellers to view the current menu.

Rationale: A seller who wishes to sell food must be able to see the currently available items and their costs to decide based on the availability of resources of the seller for either changing the menu or updating it.

2.2.2.2 Adding Menu Item

Description: This function allows sellers to add an item to the current menu.

Rationale: Whenever a seller wants to include new foods to their menu their needs to be a functionality to add an item to the menu card.

2.2.2.3 Deleting Menu Item

Description: This function allows sellers to delete an item in the current menu.

Rationale: Whenever a seller wants to remove an item from the menu there needs to be a functionality to remove an item from the menu card.

2.2.2.4 Updating Menu Item's Price

Description: This function allows sellers to update the price of a given in the menu.

Rationale: Whenever a seller wants to change the price of a menu item there needs to be a functionality to update the price of an item from the menu card.

2.2.2.5 Updating Menu Item's Quantity

Description: This function allows sellers to update the quantity of a given in the menu.

Rationale: Whenever a seller wants to change the quantity of a menu item there needs to be a functionality to update the quantity of an item from the menu card.

2.2.2.6 View all Orders

Description: This function allows sellers to view all outgoing orders.

Rationale: The sellers need to be able to view all the current orders to prepare the meals for the users.

2.2.2.7 View all Users

Description: This function allows sellers to view all the users with outgoing orders.

Rationale: This allows the sellers to identify the customers who have bought their food products, enabling them to use this data for promotional purposes and to enhance their business growth.

2.2.2.8 View all Feedback

Description: This function allows sellers to view all the feedback for each order from a user.

Rationale: Using this feedback, sellers gather insights into user experiences, aiding in product enhancements and bolstering customer satisfaction.

2.2.2.9 Remove Completed Order

Description: This function allows sellers to remove an outgoing order once the order has been completed.

Rationale: Once the seller has completed a given user's order there needs to be a functionality to remove the given order from the list.

2.2.3 User Features

2.2.3.1 Viewing Menu Card

Description: This function allows users to view the current menu.

Rationale: A user who wishes to order food must be able to see the currently available items and their costs to decide based on the cost and quantity of that menu item.

2.2.3.2 Ordering an Item

Description: This function allows users to add a given menu item giving item a name, seller name, and item amount.

Rationale: Once a user has decided on what food to order they can use this functionality to order an item into the cart where they can then proceed to place the order.

2.2.3.3 Deleting an Item

Description: This function allows users to remove a given menu item giving the item name and the seller's name.

Rationale: If a user has decided to not order and wishes to remove the items from their cart, they can use this functionality to perform the required task.

2.2.3.4 Updating Item Quantity

Description: This function allows users to update the number of a given menu item giving the item name and seller's name.

Rationale: In the case the user has decided that they require either more or less food from the seller they can use this functionality to perform the required task.

2.2.3.5 Viewing Cart

Description: This function allows users to view the current state of their cart.

Rationale: This allows the user to view all the items in their cart and make informed decisions on if any changes are required or if they can proceed to place the order.

2.2.3.6 Placing an Order

Description: This function allows the user to place an order and send it to the seller.

Rationale: Once the user has finalized his order and wishes to place it they can proceed with this, providing the name of the seller.

2.2.3.7 Providing Feedback

Description: This function allows the user to provide feedback for a given seller.

Rationale: The user can express their opinions on the food provided by the seller this allows the seller to gauge opinion and make decisions based on the feedback given by the user.

2.2.3.8 View Order History

Description: This function allows the user to view their order history.

Rationale: The order history page gives users a convenient overview of their previous orders, allows them to access specific order details, and keeps them informed.

2.3 User Classes and Characteristics

The main actors in the system are (1) the user, (2) the seller, (3) the menu card, (4) the menu item, (5) the user item, (6) the cart, and (7) feedback. The user will select a menu item from the menu card and add it to their cart, place the order, and send feedback. The seller satisfies the given order and will view the feedback:

- User ○ Has properties like username, password, order history ○ Associated with Feedback and Cart.
- Seller ○ Has properties like username, password ○ Associated with Menu card, Users, Cart, and Feedback
- Menu Card ○ Associated Menu Item
- Menu Item ○ Has properties like name, quantity, price, and seller name
- User Item ○ Has properties like name, quantity, price, and seller name
- Cart ○ Has properties like total price, username, seller name ○ Associated with User Item
- Feedback ○ Has properties like name, admin name, feedback

- Users have one to many relationships with Cart and Feedback.
- The seller has one to many relationships with the User, Cart, and Feedback.
- Menu Card has a one-to-many relationship with Menu Item.
- The cart has a one-to-many relationship with the User Item.

3. Features

This section provides detailed requirements for the website design, including functional requirements.

3.1 General Requirements

3.1.1 System Requirements

3.1.1.1 Login:	
Description:	The login access feature is integral to the functionality of the food ordering system, serving as how users and sellers authenticate themselves. When a user provides their username and password, the system undertakes a process of validation by querying its database of registered users or sellers, depending on the nature of the login request. Upon locating the provided username within the respective database, the system retrieves the associated password and compares it with the password provided by the user. Should the provided password match the retrieved password, access to the application is granted, allowing the user or seller to enter the platform. However, in the event of a mismatch between the provided and retrieved passwords, or an incorrect username, the system generates an error message indicating the discrepancy. Subsequently, the user or seller is prompted to re-enter their username and password, thereby initiating the authentication process anew.
Inputs:	Username and password
Source:	Database containing the username and password
Outputs:	An indication that the user is logged in to the system
Destination:	The outputs are displayed on the screen
Requires:	The user/seller provides login information including username and password.
Pre-Conditions:	The user is not logged into the system but is already registered.
Post-Conditions:	The user is logged in to the system.
Side-Effects:	None

3.1.1.2 Register	
Description:	The registration access feature is pivotal for authenticating users and sellers in the food ordering system. Upon submission of a username and password, the system queries its database for existing usernames. If a match is found, indicating an already registered profile, users are prompted to select a different username to avoid duplication. Conversely, if no match is found, the system proceeds to create a new record in the database, granting access to the individual. This process ensures database integrity and security by preventing duplicate entries and maintaining accurate user records.
Inputs:	Username and password
Source:	Database containing the username and password
Outputs:	An indication that the user is registered into the system
Destination:	The outputs are displayed on the screen
Requires:	The user/seller provides a username and password.
Pre-Conditions:	The user is not registered in the system.
Post-Conditions:	The user is registered in the system.
Side-Effects:	None

3.1.3: Seller Features:

3.1.2.1: Viewing Menu Card	
Description:	The seller can view their menu card within the system. If the menu card has not been created or if no items have been added to it, the seller will receive a notification indicating this status. Otherwise, the system accesses the menu card array associated with the seller and displays its contents in a structured manner. The displayed list includes the name of each menu item, its corresponding price, and the cost of the item. This presentation ensures clarity and organization, facilitating the seller's comprehension of their menu card contents.
Inputs:	Seller information – the seller must already be logged in.

Source:	Inputs are from the seller menu items array, which is retrieved from the system.
Output:	The list of all menu items given by the seller contains the name, quantity, and price of each item.
Destination:	The outputs are displayed on the screen in a structured layout.
Pre-Conditions:	The seller must be logged in.
Post-Conditions:	Sellers can access their menu cards to review current items and contemplate updates or deletions.
Side-Effects:	None.

3.1.2.2 Adding a menu item	
Description:	When the seller initiates the process to add a menu item, the system prompts them to input the food product's name, quantity, and price. Following this, the system proceeds to verify the existence of a menu card associated with the seller. If no menu card is found, the system takes the necessary steps to create one. Upon confirming the presence of a menu card or after its creation, the system utilizes the provided arguments to generate a new menu item. This menu item, comprising the name, quantity, and price details provided by the seller, is then appended to the seller's menu card. Upon successful addition, the system displays a confirmation message, notifying the seller that the item has been successfully added to their menu card.
Inputs:	Seller information – the seller must already be logged in—the name, quantity, and price of the food product.
Source:	The system checks the seller’s menu card and adds the menu item.
Output:	A success message than the food product is added to the menu card.

Destination:	The success message is displayed on the screen.
Pre-Conditions:	The seller must be logged in.
Post-Conditions:	The seller has his new food product on his menu card which users can buy.
Side-Effects:	None.

3.1.2.3 Deleting a menu item

Description:	When the seller initiates the deletion process for a menu item, they are prompted to input the food product's name. The system verifies the existence of the seller's menu card. If none exists, an error message indicates the absence of a menu card. Upon confirmation of the menu card's presence, the system checks for the specified item. If not found, an error message states that the given food name is not present. If the item is found, it is removed from the menu card, and a success message confirms the deletion.
Inputs:	Seller information – the seller must already be logged in—the name of the food product.
Source:	The system checks the seller’s menu card and deletes the menu item if found.
Output:	A success message than the food product is deleted from the menu card. Or an error message if the menu card is not yet created or if the given food name is not present on the menu card.
Destination:	The success and error messages are displayed on the screen.
Pre-Conditions:	The seller must be logged in.

Post-Conditions:	Suppose the seller gives an existing food item in his menu card. The seller will be able to delete that item.
Side-Effects:	None.

3.1.2.4 Updating a menu item's price	
Description:	When the seller initiates the updating process for a menu item's price, they are prompted to input the food product's name. The system verifies the existence of the seller's menu card. If none exists, an error message indicates the absence of a menu card. Upon confirmation of the menu card's presence, the system checks for the specified item. If not found, an error message states that the given food name is not present. If the item is found, the price of the specified menu item is updated, and a success message confirms the updating process.
Inputs:	Seller information – the seller must already be logged in—the name of the food product.
Source:	The system checks the seller's menu card and updates the menu item's price if found.
Output:	A success message than the food product's price is updated in the menu card. Or an error message if the menu card is not yet created or if the given food name is not present on the menu card.
Destination:	The success and error messages are displayed on the screen.
Pre-Conditions:	The seller must be logged in.
Post-Conditions:	Suppose the seller gives an existing food item in his menu card. The seller will be able to update that item
Side-Effects:	None.

3.1.2.5: Updating a menu item's quantity	
Description:	When the seller initiates the updating process for a menu item's quantity, they are prompted to input the food product's name. The system verifies the existence of the seller's menu card. If none exists, an error message indicates the absence of a menu card. Upon confirmation of the menu card's presence, the system checks for the specified item. If not found, an error message states that the given food name is not present. If the item is found, the quantity of the specified menu item is updated, and a success message confirms the updating process.
Inputs:	Seller information – the seller must already be logged in—the name of the food product.
Source:	The system checks the seller's menu card and updates the menu item's quantity if found.
Output:	A success message than the food product's quantity is updated in the menu card. Or an error message if the menu card is not yet created or if the given food name is not present on the menu card.
Destination:	The success and error messages are displayed on the screen.
Pre-Conditions:	The seller must be logged in.
Post-Conditions:	Suppose the seller gives an existing food item in his menu card. The seller will be able to update that item's price.
Side-Effects:	None.

3.1.2.6: To view all users	
Description:	Sellers have access to information about users who have purchased their food products. When a user buys an item from a seller, the user's name is stored in an array specific to that seller. In cases where no users are recorded, the function returns a message indicating the absence of any orders. This allows the seller to identify the customers who have bought their food products, enabling them to use this data for promotional purposes and to enhance their business growth.
Inputs:	Seller information – the seller must already be logged in.

Source:	The system checks the users' array.
Output:	A structured layout of the users who have bought their product and if no users have bought a product yet will be returned a message stating that.
Destination:	The structured layout of the users or error message is displayed on the screen.
Pre-Conditions:	The seller must be logged in.
Post-Conditions:	The seller can look at the user's name who has bought their food product.
Side-Effects:	None.

3.1.2.7: To view all orders	
Description:	Sellers have access to view orders placed with them, stored in an order array within their database. If the array is empty, a message informs the seller that no orders have been placed. However, if orders exist, each order is presented to the seller, containing details such as the purchaser's identity, the items ordered, their respective quantities, and the total bill amount. This information is presented in a structured layout.
Inputs:	Seller information – the seller must already be logged in.
Source:	The system checks the orders' array.
Output:	A structured layout of the orders placed by the user mentions the user's name and the name of the food item, with its quantity and total bill amount.
Destination:	The structured layout of the orders with the user's name, food items ordered with their quantity, and total bill amount if the order exists or an error message is displayed on the screen.
Pre-Conditions:	The seller must be logged in.

Post-Conditions:	The seller can look at the unfinished orders assigned to them.
Side-Effects:	None.

3.1.2.8: To view all feedback:	
Description:	Sellers can access feedback from users about their products. This function retrieves feedback from users who have made purchases. Feedback consists of user-provided descriptions along with their names. If no feedback exists, a message indicates "no feedback given yet"; otherwise, a structured layout displays the feedback alongside the user's name. Through this feedback, sellers gather insights into user experiences, aiding in product enhancements and bolstering customer satisfaction.
Inputs:	Seller information – the seller must already be logged in.
Source:	The system checks the users' array.
Output:	A structured layout of the feedback placed by the users mentions the user's name and the feedback provided by them.
Destination:	The structured layout of the orders with the user's name and feedback given by the user if any feedback is provided by any user to the seller otherwise, a message stating no feedback given yet.
Pre-Conditions:	The seller must be logged in.
Post-Conditions:	The seller can look at feedback provided by the users to them.
Side-Effects:	None.

3.1.2.9: To remove the finished order:	
Description:	Sellers can remove completed orders from their list by providing the user's name associated with the finished order. The system searches the seller's order array for the specified order. If the array is empty, an error message states "No orders have been placed yet." Otherwise, the order matching the user's name is deleted from the array. This process ensures efficient management of orders and allows sellers to maintain an organized list of active transactions.
Inputs:	Seller information – the seller must already be logged in.
Source:	The system checks the orders' array.
Output:	If the order is present the order is removed from the array else an error message is displayed mentioning the order is not placed yet.
Destination:	An error message about the order not being placed if the order is not found in the orders' array.
Pre-Conditions:	The seller must be logged in.
Post-Conditions:	The seller can be able to delete the completed order from the orders' array.
Side-Effects:	None.

3.1.3: User Features:

3.1.3.1: View Menu Card:	
Description:	The user can view their menu card within the system. The system accesses the menu card array associated with the seller and displays its contents in a structured manner. The displayed list includes the name of each menu item, its corresponding price, and the cost of the item. This presentation ensures clarity and organization, facilitating the user's comprehension of their menu card contents.
Inputs:	User information – the user must already be logged in. Seller information – the sellers must exist to view the menu card.

Source:	Inputs are from the seller menu items array, which is retrieved from the system.
Output:	The list of all menu items given by the seller contains the name, quantity, and price of each item.
Destination:	The outputs are displayed on the screen in a structured layout.
Pre-Conditions:	The User must be logged in.
Post-Conditions:	Users can access their menu cards to review current items.
Side-Effects:	None.

3.1.3.2: Ordering an Item:	
Description:	When a user searches for a specific item, the system displays a list of all the sellers who are currently selling that item, if any. If there are no sellers available, the system will display a message indicating that there are no sellers for that item. If the item is available, the user can proceed by selecting a seller and entering the desired quantity of the item. If the quantity of items required by the user exceeds the quantity of items available for the seller, the system will display an error message. Otherwise, the item will be added to the user's cart.
Inputs:	User information – the user must already be logged in. Seller information – the sellers must exist to show the available items.
Source:	Inputs are from the seller menu items array, which is retrieved from the system.
Output:	A structured layout of the orders placed by the user mentions the seller's name and the name of the food item, with its quantity and total bill amount.
Destination:	The structured layout of the orders with the user's name, food items ordered with their quantity, and total bill amount if the order exists or an error message is displayed on the screen.
Pre-Conditions:	The user must be logged in.
Post-Conditions:	The user can look at the ordered Items in the ordered History.
Side-Effects:	None.

3.1.2.3 Deleting an item	
Description:	When the user initiates the deletion process for a menu item, they are prompted to input the food product's name. The system verifies the existence of the user's cart. If none exists, an error message indicates the absence of a user's cart. Upon confirmation of the user's cart presence, the system checks for the specified item. If not found, an error message states that the given food name is not present. If the item is found, it is removed from the menu card, and a success message confirms the deletion.
Inputs:	User information – the user must already be logged in—the name of the food product.
Source:	The system checks the user's cart and deletes the menu item if found.
Output:	A success message than the food product is deleted from the menu card. Or an error message if the menu card is not yet created or if the given food name is not present on the menu card.
Destination:	The success and error messages are displayed on the screen.
Pre-Conditions:	The user must be logged in.
Post-Conditions:	Suppose the user gives an existing food item in his user cart. The user will be able to delete that item.
Side-Effects:	None.

3.1.3.4: Updating an item's Quantity:	
Description:	When the user initiates the updating process for a menu item's quantity, they are prompted to input the food product's name. The system verifies the existence of the User's cart. If none exists, an error message indicates the absence of a menu card. Upon confirmation of the user's cart presence, the system checks for the specified item. If not found, an error message states that the given food name is not present.

	If the item is found, the quantity of the specified menu item is updated, and a success message confirms the updating process.
Inputs:	User information – the user must already be logged in—the name of the food product.
Source:	The system checks the user's cart and updates the menu item's quantity if found.
Output:	A success message that the food product's quantity is updated in the user's cart. Or an error message if the user's cart is not yet created or if the given food name is not present on the user's cart.
Destination:	The success and error messages are displayed on the screen.
Pre-Conditions:	The user must be logged in.
Post-Conditions:	If the user gives an existing food item in his user's cart. The user will be able to update that item's quantity.
Side-Effects:	None.

3.1.3.5: Viewing the Cart:	
Description:	Users can access their respective cart array, which is stored in an order array in their database. If the cart array is empty, a message will inform the user that no items have been added to the cart. However, if there are items in the cart, each item will be displayed to the user along with details such as the identity of the purchaser, the name of the corresponding seller, the ordered items, their respective quantities, and the total bill amount. The information will be presented in a structured layout for easy understanding.
Inputs:	User information – the user must already be logged in.
Source:	The system checks the user's cart array.
Output:	A structured layout of the items placed by the user mentions the user name and the name of the food item, with its quantity and total bill amount.
Destination:	A structured layout of the items placed by the user mentions the user name and the name of the food item, with its quantity and total bill

	amount if the item exists or an error message is displayed on the screen.
Pre-Conditions:	The user must be logged in.
Post-Conditions:	The user can look for what items have been included.
Side-Effects:	None.

3.1.3.6: Placing an Order:	
Description:	Users can place an order by entering the name of the seller from whom they wish to place the respective orders. If the user's cart is empty, an error message will be displayed stating "Cart is empty". However, if there are items in the cart, the orders can be placed with the given seller's name. Users can place an order by entering the seller's name to which the respective orders as to be placed.
Inputs:	User information – the user must already be logged in.
Source:	The system checks the user's cart array.
Output:	The success and error messages are displayed on the screen.
Destination:	The success and error messages are displayed on the screen.
Pre-Conditions:	The user must be logged in.
Side-Effects:	None.

3.1.3.7: Providing Feedback	
Description:	Users can provide feedback by entering the name of the seller if the seller is available. If the seller is not available, the user will be notified accordingly. If the seller is found, the user can enter their feedback as required and proceed to send it to the respective seller.
Inputs:	User information – the user must already be logged in.
Source:	The system checks the users' array.
Output:	The success and error messages are displayed on the screen.
Destination:	The success and error messages are displayed on the screen.
Pre-Conditions:	The user must be logged in.
Side-Effects:	None.

3.1.3.7: Viewing Order History:	
Description:	Sellers have access to view orders placed with them, stored in an order array within their database. If the array is empty, a message informs the seller that no orders have been placed. However, if orders exist, each order is presented to the seller, containing details such as the purchaser's identity, the items ordered, their respective quantities, and the total bill amount. This information is presented in a structured layout.
Inputs:	User information – the user must already be logged in.
Source:	The system checks the orders' array of Users.
Output:	A structured layout of the orders placed by the user mentions the seller's name, the user's name, and the name of the food item, with its quantity and total bill amount.

Destination:	The structured layout of the orders with the user's name, seller's name, food items ordered with their quantity, and total bill amount if the order exists or an error message is displayed on the screen.
Pre-Conditions:	The user must be logged in.
Post-Conditions:	The user can view their previously ordered items.
Side-Effects:	None.

4. External Interface Requirements

4.1 User Interfaces

A User-Friendly Interface for the users to order food, and submit feedback. The interface is also made engaging for the food dealers to track their orders, and menu cards and validate the feedback received from them.

4.2 Communications Interfaces

The system must utilize the standard Hypertext Transfer Protocol (HTTP) to ensure maximum inter-browser compatibility. The client accesses the system through a web browser.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The software uses Next JS over React which provides excellent Typescript support that improves the user experience.
- Next.js automatically splits JavaScript bundles based on page boundaries, enabling efficient loading of only the necessary code for each page. This helps in reducing initial loading times and improving overall performance.

5.2 Security Requirements

- Passwords must be a minimum of eight characters and must contain one to seven digits.

- Email addresses should be verified before the system grants user access. The username in the Systems Design component is designed in such a way that no two same usernames can exist which provides some sort of minimal security.

- All exchanges from client to server involving private data shall occur using the highest available level of secure connection (e.g., HTTP).

5.3 Software Quality Attributes

5.3.1 Usability:

The Food Ordering website design shall allow deployment on both Windows and UNIX (Linux) servers. The design should support Windows Server 2003, and Linux 2.6.x, V10 UNIX, and later.

5.3.2 Robustness:

Implemented robust error handling mechanisms to gracefully handle errors display the error messages to users and ensure that the system can recover from errors without losing critical data or disrupting ongoing orders.

6. Appendix A: Glossary

NEXT.Js	Next.js is a React framework that enables server-side rendering, automatic code splitting, and simplified routing for building fast, scalable web applications.
TypeScript	TypeScript is a statically typed superset of JavaScript that compiles to plain JavaScript providing improved tooling, code maintainability, and scalability for web development projects.
Database	A structured collection of data organized for efficient retrieval, storage, and Manipulation.
SQL	Database query language for managing and manipulating relational databases
HTTP	Protocol for communication between web servers and clients.
CSS	Styling language for defining the presentation and layout of HTML documents.
Input Criteria	Specifications or conditions used to define acceptable inputs for a system or process.