|  |  |
| --- | --- |
| Slide | Content |
| 1 | Hi, very Good Morning to you all. Today we are going to present about the web application built for Food Cone restuarent. |
| 2. | These are the main content that we are going to cover up today. Without further delay lets move on to the introduction. |
| 3 | Food Cone web application is designed for a restuarent to market their food items among a vast majority of people and provide their customers a more reliable and easy way to identify their items and purchase them.  the whole project relies on the concept of Shopping cart design where a user can log in to the system, he she will have a own cart and he or she can purchase any number of items and do the check out finally. This method is used in e-commerce platforms and considered as a more suitable way to implement product selling applications. When developing this project MongoDb, Express, React and Node is used which is known as MERN. Explanations about the development will be done in the later part of the presentation. AWS has been used for the hosting of the application. lets move on to the main system functionalities. |
| 4 | As the main functionalities of the user user can view items, change the quantity he is ordering, add items to cart are implemented. Also when we buy something we always consider the feedback of previous buyers. So providing feedback and viewing feedbacks is also implemented in this system. as a additional feature users can chat with the restuarent using the chat bot. |
| 5 | As for admin users basically the owner and the workers of the restuarent can perform actions like add, update and remove items from the system, chat with customers, view feed back and manage users. Next lets move on to the basic concepts of web application development using MERN. |
| 6 | When we consider the basic concepts these 7 categories can be taken as the main basic concepts of developing a web application using mern. Lert me explain each. The MERN stack is composed of four key technologies: MongoDB, Express.js, React.js, and Node.js. Each of these components plays a crucial role in building a web application. MongoDB is our database, Express.js and Node.js form the server-side, and React.js is used for the client-side. then when we talk about the Frint end and the back end The frontend is built using React.js, which helps create a dynamic and interactive user interface. On the other hand, the backend is managed by Node.js and Express.js. Node.js handles the server-side logic, while Express.js is used to create APIs and handle routing. when the UI and the API is complete we have to consider about the database which we used MongoDB as a no sql database. It allows for efficient and flexible storage of data in a JSON-like format, making it ideal for modern web applications where the data structure can vary. upnext you may wonder how these separate parts communicate with each other to perform actions. It is handled through RESTful APIs. Express.js is used to create these APIs, which allow the frontend to send requests to the backend and receive data asynchronously. |
| 7 | Asynchronous Operations,Asynchronous programming is essential in web development to handle operations like data fetching without blocking the main execution thread. In the MERN stack, we use Promises and Async/Await to manage these asynchronous operations efficiently.  Next State Management. Managing the state of the application is crucial, and React provides tools like Hooks and the Context API for this purpose. These tools help manage and share state across different components in a predictable and efficient manner.  Last Component-Based Architecture. React's component-based architecture allows us to build reusable UI components. This approach promotes modularity and maintainability, as each component can be developed, tested, and maintained independently. up next lets move on to a more detailed explanation of the data base design. |
| 8 | When we move on to the database design the first question comes is Why Use MongoDB?. MongoDB is a NoSQL database, which means it does not rely on a fixed schema like traditional relational databases. This flexibility allows for easy modifications to the data structure as our application requirements change. MongoDB is also designed for scalability and performance, making it suitable for applications that need to handle large amounts of data and high traffic.  Next move on to the advantages of using mnongoDB  High Availability with Replication -MongoDB's replication feature ensures data is copied across multiple servers, providing redundancy and high availability.  Horizontal Scalability with Sharding - Sharding allows us to distribute data across multiple machines, supporting the growth of our application as data volumes increase.  Document-Oriented Storage-Data is stored in JSON-like documents, which makes it easier to manage and retrieve complex data structures.  Ease of Use-The JSON-like format of documents is both human-readable and easy to manipulate programmatically, streamlining the development process.  So what are the implimentations using MongoDB in our system ?  Storing Restaurant Menu Items is the main fact of using MongoDb. Each menu item, with its details, is stored as a document, allowing for easy updates and retrieval.  Orders and reservations are tracked efficiently, enabling real-time updates and seamless management.  User Authentication and Profiles meansUser data is securely stored and managed, ensuring a smooth and secure user experience.  And also MongoDB supports real-time data operations, providing valuable insights into customer behavior and restaurant operations. |
| 9 | This app was created primarily using MERN (Mongo , Express , React , Node) to both client-side and serverside programming. The frontend of the food delivery web app is built using React.js. This is the part of the application that users interact with directly. It handles the user interface, user interactions, and communicates with the server to fetch and update data.  The user interface supports fUser interactions, form submissions, and navigation are handled through React components. Components utilize state hooks (useState) for managing local component-level state. The client communicates with the server using the Fetch API for making HTTP requests. API calls are asynchronous and handled in functions such as handleSubmit in the user login and food ordering page.React Router (useNavigate and Link from 'react-router-dom') is used for client-side navigation between different pages.The useCart and useDispatchCart hooks suggest the usage of React Context API for managing global state related to the shopping cart. |
| 10 | The server is implemented using Node.js with Express.js as the web application framework, which offers a server-side runtime environment and a framework for constructing  APIs, is used for the back end. Together, these components make up the server-side architecture, which manages data storage and communication with the clientside. Scalability  and effectiveness are provided via the event-driven, non-blocking architecture of Node.js MongoDB is used as the database for storing user data, reviews, and order information. I picked MongoDB for the database because, it saves data as documents in a JSON format. It effectively keeps track of files, shared information, and received content. The Mongoose library, an ObjectDocument Mapping (ODM) library for MongoDB and Node.js, is used to retrieve the data in this project, which is being stored in the MongoDB database. By bridging the gap between the client and the database, the API layer enables data retrieval and editing. The application adds a layer of interaction through the usage of WebSocket technology. Instant feedback to users is made by this technology's facilitation of real-time communication between the client and server. |
| 11 |  |