Frontend Developer

Bhargava Teja Narasani

Email: nbhargavteja2022@gmail.com
Mobile: +91-9036400498

LinkedIn **Portfolio** GitHub Repository

Carrier Objective:

- To attain deepest knowledge and skills in the field of interest and to render excellent service in a company that offers professional growth while being resourceful, innovative and flexible.

Positive Skills

- Ability to work in team dedicated environment with professional attitude.
- Effective communication, presentation skills and hardworking nature.
- Comprehensive problem-solving skills, motivated and quick learner.
- Ability to articulate and compare alternative approaches.

Work Experience

- Total work Experience: 1+yrs
- Relevant to Commvault: 1+yrs
- Worked as Software Engineer in **TEMENOS** from Dec 2021 to Jan 2023

Academic Details

- B. Tech – Vignan University - ECE – 2017-2021 - 8.2 CGPA
- 12th Standard – Narayana Junior College – 2015-2017 – 91%
- Schooling – Shree KVR Naidu High School – 2014-2015 – 8.0 CGPA

Technical Skills

- Proficient in React.js, Nextjs, Redux-Toolkit, Javascript & TypeScript
- Experience with REST/CONTEXT-API, Node.js, Express.js, Mongo DB, Firebase, Python, SQLite, HTML, CSS
- Familiarity with Bootstrap and Material UI frameworks and React-Native
- Knowledge of version control systems such as Git
- Experience with tools like Postman, Jira, and Bitbucket

Roles and Responsibilities:

- Developing and implementing user-friendly websites and applications using ReactJS, Nextjs, Redux-Toolkit, Javascript, Typescript, HTML and CSS.
- Using Node.js and Express.js to build RESTful APIs for frontend applications to consume
- Working with databases such as MongoDB, Firebase to store and retrieve data
- Optimizing the performance of web applications through techniques such as code splitting, lazy loading, and caching

- Participating in Agile development methodologies such as Scrum or Kanban, attending daily stand-ups, sprint planning, and retrospective meetings
- Mentoring junior developers and providing guidance on coding best practices and development workflows.
- Collaborating with cross-functional teams, including designers, developers, and project managers, to deliver high-quality projects on time and within budget.
- Worked closely with clients in order to understand and solve their requirements in fast paced manner
- Keeping up-to-date with the latest technologies, trends, and frameworks in web development, and applying them to enhance the performance and functionality of projects
- Writing and maintaining documentation for projects, including technical specifications, user manuals, and design guidelines
- Collaborating with backend developers to ensure efficient and reliable data flow
- Communicating effectively with team members and stakeholders to ensure successful project delivery
- Implementing responsive web design and ensuring cross-browser compatibility

Projects:

Threads Recreated: A Clone That Resembles Innovation (Click Here)

Project Description: The Threads Clone is a dynamic and user-centric platform built using cutting-edge technologies to facilitate engaging discussions and community interactions. It offers features like thread creation, commenting, community formation, user notifications, and seamless social logins via Google and GitHub.

Technologies Used: Next.js 13.4, TypeScript, MongoDB & Mongoose, Tailwind CSS, shadcn/ui, React Hook Form, Zod, Clerk, HTML, CSS, JavaScript

Key Features Implemented:

- Thread Creation and Commenting: Users can create and post threads, as well as comment on their own or others' threads, fostering meaningful conversations within the community.
- Community Formation: Users can establish communities and invite members to join. Threads can be posted within communities, enhancing organization and fostering niche discussions.
- User Notifications: The "Activity" tab provides notifications for user interactions, such as receiving comments on their threads, ensuring users stay engaged with their content.
- Search Functionality: A search page allows users to find different people and threads based on various criteria, enhancing discoverability and user engagement.
- User Profile: The profile page showcases user details, including bio, profile photo, and threads created. Users can manage and delete their threads from this page.
- Smooth Media Uploads: The Upload Thing feature enables seamless media uploads, enriching thread content and user engagement.
- Authentication with Social Logins: Integration of Clerk enables users to log in using Google or GitHub accounts, streamlining the authentication process.

Challenges Faced:

- Responsive UI Design: Designing a responsive interface that works seamlessly across various devices and screen sizes to provide consistent user experiences.
- TypeScript Integration: Integrating TypeScript with Next.js and Tailwind CSS to enhance code reliability, maintainability, and development efficiency.
- Data Management: Efficiently managing data from various sources, including MongoDB and external APIs, while ensuring proper error handling.
- Dynamic Content Loading: Implementing dynamic loading of threads and comments to optimize performance while maintaining a smooth user experience.

Solutions Implemented:

- Next.js Framework: Utilizing Next.js to create a responsive design that adapts to different devices, leveraging server-side rendering for improved performance.
- Tailwind CSS: Leveraging Tailwind CSS for rapid styling, ensuring consistent design across the application while saving development time.
- TypeScript Benefits: Incorporating TypeScript to detect bugs early, enhance code readability, and provide a better development experience.
- Headless UI Integration: Utilizing the Headless UI library for pre-built components like dropdowns and autocomplete search, enhancing user interaction and reducing development time.
- Efficient Data Handling: Implementing data fetching and pagination techniques to optimize content loading, ensuring an optimal user experience.

POKEMON (Click Here for the Preview)

Project Description: The Pokémon Application is a feature-rich platform developed using React, Redux Toolkit, TypeScript, and Firebase to provide an engaging and comprehensive Pokémon experience. This application offers users the ability to search, view details, compare, and manage their favorite Pokémon, leveraging Firebase for authentication and data storage.

Technologies Used: React, Redux Toolkit, TypeScript, Firebase

Key Features Implemented:

- Search and Details View: Users can search for specific Pokémon, click on them to access detailed information, and view evaluations, creating an immersive experience.
- Pokémon Comparison: The application enables users to compare the attributes of two Pokémon side by side, aiding in strategic decisions and analysis.
- Favorite Pokémon List: Users can create a list of their favorite Pokémon by adding specific ones to their favorites, ensuring easy access and personalization.
- Authentication with Firebase: Firebase is integrated for user authentication, enhancing security and enabling personalized experiences for authenticated users.
- State Management with Redux Toolkit: Redux Toolkit is employed to manage the application's state, allowing consistent and efficient data flow across the entire application.

Challenges Faced:

- Data Management: Efficiently fetching and storing Pokémon data from external sources while ensuring smooth interactions and updates in the application.
- Integration with Firebase: Configuring Firebase for authentication and data storage, including setting up user-specific favorite lists.
- State Management Complexity: Handling the application's state using Redux Toolkit while maintaining code simplicity and readability.
- User Experience: Designing a user-friendly interface that allows users to seamlessly search for, compare, and manage their favorite Pokémon.

Solutions Implemented:

- External API Integration: Efficiently fetching Pokémon data from external APIs and storing it in the application's Redux store for easy access and interaction.
- Firebase Authentication: Utilizing Firebase authentication to secure user accounts and manage authentication-related features.
- State Management Streamlining: Using Redux Toolkit's built-in tools for state management, including slices and reducers, to simplify state management and maintain a clean codebase.
- User-Centric Design: Designing an intuitive and user-friendly interface that enhances user engagement and encourages exploration of the application's features.

Prompt Me - Empowering Creativity (Click Here for the Preview)

Project Description:
PromptMe is a web application built using Next.js, Tailwind CSS, and NextAuth, designed to inspire creativity and encourage users to express their thoughts through writing prompts. The app provides a platform for users to sign in with their Google accounts, create writing prompts, and view & copy them in a centralized home page.
Users can also access their profile, where they can manage and edit their created prompts. The app supports searching for prompts based on title, content, and tags, making it easy for users to find inspiration for their writing endeavors.

Technologies Used:

Next.js: A popular React framework for server-rendered React applications.
Tailwind CSS: A utility-first CSS framework for rapidly building custom user interfaces.
NextAuth: A library for authentication in Next.js applications that supports various providers, including Google.
Google Auth: Allows users to sign in with their Google accounts securely.

Key Features Implemented:

User Authentication: Integration of NextAuth and Google Auth providers for secure user sign-in and authentication.
Prompt Creation: Users can create prompts with a title, content, and tags for categorization.

Home Page: Displaying a list of all created prompts for users to explore and find inspiration.
Profile Section: Users can view their own created prompts and perform actions like editing or deleting them.
Search Functionality: Users can search for prompts based on title, content, or tags to find specific prompts easily.

Challenges Faced:

Implementing Google Auth: Integrating the Google authentication provider and handling user data securely.
Prompt Search: Designing an efficient and effective search system for prompts based on various criteria.
Managing Prompt Editing: Ensuring that users can safely edit and delete their prompts without compromising data integrity.

Solutions Implemented:

Google Auth Integration: Properly configuring NextAuth with Google Auth to enable secure user sign-in.
Optimized Search Functionality: Implemented a robust search system using Next.js and Tailwind CSS to allow users to find prompts efficiently.
Safe Prompt Management: Developed a secure backend system to handle prompt editing and deletion requests while validating user permissions.

Car Connect: Simplified Car Search Experience (Click Here for the Preview)

Project Description: The application allows users to search for cars based on various criteria such as car company, car type, fuel type (including fuel, gas, and electric), and year
Technologies Used: Next.js, TypeScript, Tailwind CSS, Headless UI, React, HTML, CSS
Key Features Implemented:

- User-friendly interface for seamless car searching and browsing experience
- Search functionality allowing users to find cars by company, type, fuel type, and year
- Integration of the Headless UI library to create dropdowns and autocomplete search for enhanced user interaction
- Displaying detailed information about a selected car in a popup box, featuring a close icon for easy dismissal
- "Show more cars" button at the bottom of the page, dynamically increasing the displayed car limit (e.g., from 10 to 20, 20 to 30) upon clicking
- Implementation of TypeScript for enhanced code reliability and maintainability

Challenges Faced:

- Designing and implementing a responsive user interface that works seamlessly across different devices and screen sizes
- Integrating and configuring TypeScript with Next.js and Tailwind CSS for type safety and improved development experience
- Managing and consuming data from the Cars API, ensuring proper handling of responses and errors
- Implementing the dynamic increase of the displayed car limit while maintaining optimal performance

Solutions Implemented:

- Utilizing Next.js, a powerful framework for building server-rendered React applications, to achieve a responsive design that adapts to various devices

- Leveraging Tailwind CSS, a utility-first CSS framework, to expedite the styling process and ensure consistent design across the application
- Utilizing the TypeScript language to catch potential bugs early, enhance code readability, and provide a better developer experience
- Integrating Headless UI to leverage pre-built components for dropdowns and autocomplete search, reducing development time and improving user interaction
- Implementing efficient data fetching and pagination techniques to optimize the loading and displaying of cars, enhancing overall performance

Achievements And Awards

- Secured Preliminary English Test (PET) certificate conducted by Cambridge with merit B1 CEFR level in 2017
- Qualified in business English certificate Vantage held by Cambridge University - 2019

Certificates

- Spot Award for Best Presentation After Compilation of Training By Temenos
- 4 Certificates on Front-End Which Was Issued by NXT WAVE.

Personal Information:

- Date of birth : 04/10/2000
- Sex : Male
- Marital status : Single
- Languages known : English, Telugu and Hindi.
- Nationality : Indian
- Strength : Goal oriented and ready to take challenge

Declaration:

I hereby declare that above written particulars are true to the best of my knowledge and belief.

Date:

Place: Bangalore (Bhargava Teja)