

# Bhargav Teja

REACT | NEXT JS DEVELOPER

[LinkedIn](#) [Portfolio](#) [GitHub Repo](#)



+91 9036400498



nbhargavteja2022@gmail.com



Bengaluru Karnataka

## SKILLS

JavaScript, TypeScript (Basic), HTML, CSS

React, Nextjs, Redux-Toolkit, HTML5,

CSS3, Tailwindcss

Node.js, Express.js

MongoDB, MySQL

Git, Github

RESTful API Development

Authentication and Authorization (JWT)

Bootstrap, Material Ui, OAuth,

Firebase

## EDUCATION

### Bachelor of Engineering

Vignan University  
Guntur, Andhra Pradesh

2017-2021 8.2/10 CGPA

### Intermediate (12th Grade)

Narayana Junior College

2015-2017 91% /100%

### SSC (10th Grade)

Sree KVR Naidu High School

2015 8.0/10 CGPA

## LANGUAGES

English Telugu

Hindi

## SUMMARY

Passionate and skilled React/NEXTJS developer (MERN) (MongoDB, Express, React, Node.js) with 1 years of experience creating dynamic and interactive web applications. Proficient in developing user interfaces with React and related technologies, I thrive on solving complex problems and delivering seamless user experiences. With a solid understanding of front-end development, state management, and modern web technologies, and I had knowledge of back-end technologies such as Node.js, Express, and database systems like MongoDB. This allows me to build comprehensive full-stack applications and communicate effectively with cross-functional teams. I am dedicated to producing high-quality code

## WORK EXPERIENCE

### Software Engineer

Temenos • Bangalore, Karnataka

12/2021 – 1/2023

- Developed and maintained full-stack web applications using the MERN(MongoDB, Express.js, React.js, Node.js) stack.
- Implemented RESTful APIs and integrated them with front-end components to enable seamless data flow. Designed and built responsive user interfaces using React.js, HTML, CSS, and JavaScript, ensuring cross-browser compatibility.
- Implemented state management with Redux to maintain application data flow and improve performance Utilized Node.js and Express.js to develop server-side logic and handle API requests and responses.
- Implemented user authentication and authorization features using technologies such as JWT (JSON Web Tokens) and bcrypt.
- Utilized MUI v5, Bootstrap to implement pre-built React components and templates, saving development time and ensuring consistent styling across the application
- Participated in agile development methodologies, including daily stand-ups, sprint planning, and retrospective meetings.
- Utilized Git for version control and collaborated with team members using collaborative workflows
- Developed RESTful APIs using Node.js and Express.js for seamless communication between the front-end and back-end.
- Designed and maintained MongoDB databases, optimizing data retrieval and storage processes.

## CERTIFICATES

React certification in Hacker rank,

Best Presentation from my office

Certification in the NXT Wave,

MERN Stack courses on Udemy

May more Certifications from Online

## PROJECTS

### Tours & Travels: ([Click Here](#))

**Project Description:** Developed a robust and user-friendly Tours and Travels application using the MERN stack, offering seamless booking and management functionalities for travelers. Implemented secure user authentication and interactive features to enhance the overall user experience.

**Technologies Used:** Reactjs, Javascript, Nodejs, ExpressJs, MongoDB & Mongoose, HTML, CSS

#### Key Features Implemented:

- A MERN Stack Application Which Allows Us to Book Tours, Comment on Different Tours it can Provide Robust, Scalable & Responsive Web Application with a smooth user experience.
- The Backend of The Application Was developed by using Nodejs, ExpressJs, Restful API endpoints for communication With the Client. MongoDB Can Be Used as the Database for Storing Customer Data, Tours Data, ETC used bcrypt and jsonwebtoken for the authentication
- Defined The Schema for the user, Tours, My Bookings and used the Mongoose Package to interact with the MongoDB database.
- We need to register with an email and login, we can see all the featured tours and normal tours.
- We can select a particular tour and book tour while providing some additional Details.
- Along With that we can provide feedback like providing rating and comment, we can edit and delete later on
- We have a separate my bookings page to show all our bookings.

### Snap Threads: ([Click Here](#))

**Project Description:** A Social Media application using the React as Frontend and Appwrite as the backend. Implemented secure user authentication and interactive features to enhance the overall user experience.

**Technologies Used:** Reactjs, React Query, Appwrite, Shadcn/UI, React Hook Form, Zod, Tailwindcss.

#### Key Features Implemented:

- A Full Stack Application Which Allows Us to Create an Account, Post pics with captions and tags. it can Provide Robust, Scalable & Responsive Web Application with a smooth user experience.
- The Backend of The Application Was developed by using Appwrite for communication With the Client. Appwrite Can Be Used as the Database for Storing Customer Data, Posts Data, ETC Used inbuilt methods for fetching the logged in user.
- Defined The Schema for the user, Posts, Saved Posts.
- We need to register with an email and login, we the same email to use the application.
- We have a dedicated profile page, explore page, People Page, Saved Posts Page, Create Post Page. Implemented infinite scrolling for the explore page
- We can like and save other people's posts as well.
- If we are the current logged in user, we can edit our posts and save it again.
- We can search for posts based on tags or name

## 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.

## 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, Javascript, HTML, CSS

### 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.

**Skills:** React.js | NEXTJS | Node.js| Express | JavaScript | HTML | CSS | Redux-Toolkit | Material UI |REST | Mongo DB | Github | Git | POSTMAN