Muhammad Tayyab

Software Engineer

Mobile: +92-336-7818585 , Email: iamtayyab336@gmail.com

GitHub: github.com/iamtayyab007

SUMMARY

Software Engineering graduate with strong skills in React and building user-friendly web applications. Has completed several self-led projects, showing good knowledge of front-end development and problem-solving

EDUCATION

BSc. Software Engineer 3.37/4.0
University of Engineering & Technology, Bannu

EXPERIENCE

MERN Stack Developer (React + Next.js + TypeScript)

Current Experience | 2024 - Present

- Developing **full-stack applications** using **MERN stack (MongoDB, Express.js, React, Next.js, Node.js)**.
- Building **high-performance, scalable, and secure web applications** with modern technologies.
- Writing **clean, maintainable, and type-safe code** using **TypeScript**.
- Integrating authentication using **JWT, NextAuth.**
- Implementing state management with **Redux Toolkit and Context API**.
- Working with **server-side rendering (SSR) and static site generation (SSG)** in Next.js.
- Ensuring **SEO optimization, performance improvements, and accessibility** best practices.

Web Developer (WordPress – SEO – Online Monetization) Self Employed

Previous Experience

- Conducted keyword research to improve website rankings and gain organic traffic.
- Performed on-page SEO optimization and content improvements.
- Identified content gaps and developed strategies to create engaging and relevant content.
- Built and monetized blogs, driving consistent high-traffic growth.

- Did website speed optimization techniques to enhance user experience and SEO performance.
- Managed end-to-end SEO strategies to increase visibility and revenue from monetization platforms like Adsense, PubGalaxy and Ezoic.

Projects

1. True Feedback

A secure and AI-powered anonymous feedback platform built with Next.js and TypeScript, allowing users to receive confidential feedback while ensuring authentication and validation.

- Developed using **Next.js 14 (App Router)** with **TypeScript** for scalability and maintainability.
- Integrated **NextAuth.js with custom credentials** for secure authentication.
- Used **Zod** for robust input validation, ensuring data integrity.
- Implemented **Resend email service** for OTP verification to authenticate users securely.
- Utilized **MongoDB Aggregation Pipeline** for efficient data processing and analytics.
- Designed an intuitive and accessible UI with **ShadCN** components for a modern user experience.
- Integrated **React Hook Form** with debouncing to optimize user input handling.
- Added an **AI-powered message suggestion feature**, enhancing user engagement with intelligent feedback recommendations.
- Ensured seamless performance with **server-side rendering (SSR) and API route optimizations** in Next.js.

2. Full-Stack Authentication in NextJs

A secure and modern authentication system built with **Next.js** and **TypeScript**, featuring user registration, login, and email verification.

- Developed using **Next.js 15** for server-side rendering (SSR) and API route handling.
- Utilized **bcryptjs** for password hashing, ensuring secure storage of user credentials.
- Implemented **jsonwebtoken (JWT)** for secure authentication and session management.
- Integrated **Mongoose** with **MongoDB** for efficient user data storage and retrieval.
- Used **Nodemailer** to send verification emails and password reset links.
- Employed **Axios** for seamless API requests and error handling on the frontend.
- Designed a user-friendly interface with smooth authentication flows.
- Implemented role-based access control (RBAC) to manage different user permissions.

3. Expense Tracker App

A digital solution for managing personal finances, enabling users to track income and expenses, categorize spending, and analyze data visually.

- Developed with the React', leveraging Context API for state management.
- Designed a clean and intuitive UI for tracking personal expenses and income.
- Implemented a responsive layout using Tailwind CSS, ensuring smooth use on various devices.
- Integrated real-time data updates and interactive pie charts for data visualization.
- Enabled efficient data handling and storage with local storage.
- Created user-friendly modals for adding and editing expense/income entries.
- Ensured a seamless user experience by implementing features for expense categorization and total balance calculation.

4. Gemini Clone

A React-based project that aims to replicate the functionality of Google Gemini's AI Chatbot. It uses the Google Gemini API to enhance conversational interaction.

- Developed using React, leveraging the Gemini API for real-time cryptocurrency data integration.

- Designed and implemented a responsive and modern user interface using Tailwind CSS.
- Managed global state and seamless data sharing across components using Context API.
- Incorporated dynamic data visualization for user-friendly market analysis and trends.

5. MERN Stack Notes App

A feature-rich digital notes application designed to help users efficiently create, organize, and manage their notes.

- Built using the MERN stack, with React for the frontend, Node.js and Express.js for the backend, and MongoDB for secure data storage.
- Designed an intuitive interface allowing users to create, update, delete, and search notes with ease.
- Styled the application for responsiveness and modern aesthetics using Tailwind CSS.
- Integrated rich text editing and categorization features for enhanced note organization.

6. MERN Stack Blog App

A dynamic and user-friendly blogging platform for creating, editing and deleting blog posts.

- Developed using the MERN stack, featuring React for the frontend, Node.js and Express.js for the backend, and MongoDB for robust data storage.
- Designed and implemented features for creating, editing, and deleting blog posts.
- Styled the application with Tailwind CSS to ensure a responsive and modern design.
- Incorporated a rich text editor for content formatting and commenting functionality to enhance user engagement.

7. Login and Registration Form with Authentication System

A user authentication system allowing users to securely register, log in, and access protected content.

- Developed using the MERN stack, with React for the frontend, Express.js for the backend, and MongoDB for data storage.
- Created registration and login forms that collect user details (username, email, password) and validate inputs.
- Implemented user authentication with JWT (JSON Web Tokens) for secure session management and tokenbased authentication.
- Utilized bcrypt for password hashing to ensure secure storage of user credentials in MongoDB.
- Set up validation and error handling for both frontend and backend to ensure the integrity and security of the authentication process.
- Used React Router for navigation and redirection, ensuring users are redirected to appropriate pages based on their authentication status.
- Enabled user login via local email/password, and implemented user session handling using JWTs stored in cookies for state persistence.

8. Shopping Cart App

An advanced e-commerce platform enabling users to browse, select, and manage products with a seamless shopping experience.

- Developed using React, integrating React and Redux Toolkit for state management.
- Fetched product data dynamically from an external API.
- Implemented a robust shopping cart system with features to add, remove, and update the product.
- Utilized Redux Toolkit for efficient global state management, ensuring scalability and performance.
- Styled for responsiveness and a modern user interface using Tailwind CSS.

9. Food Recipe App

An intuitive application for discovering and exploring a variety of food recipes, catering to users' culinary interests.

- Built using React and Context API for efficient state management across components.
- Integrated a third-party API to fetch and display a wide range of recipes dynamically.
- Implemented search and filter functionalities to help users find recipes by ingredients, cuisine, or dietary preferences.
- Designed a clean and responsive user interface using Tailwind CSS for modern aesthetics.
- Enhanced user experience with loading states, error handling, and a recipe detail view showcasing ingredients and step-by-step instructions.

10. AI Image Generator

An AI-powered image generator that creates custom images based on user-provided text prompts using OpenAI's API.

- Developed using React for the frontend, providing an intuitive interface for users to enter text prompts and generate images.
- Integrated OpenAI’s API to process text inputs and generate images based on user descriptions.
- Set up state management with React hooks (useState, useEffect) to handle user inputs and display generated images.
- Implemented loading and error handling mechanisms to ensure a seamless user experience.
- Used fetch for making API requests to OpenAI's image generation service.
- Created a user-friendly interface that allows users to download and share the generated images.

11. Currency Converter with Custom Hook

A user-friendly currency converter application designed to convert between different currencies in realtime using an external API.

- Built with React and designed for seamless user experience in converting currencies.
- Created a custom React hook to handle fetching exchange rates and converting between selected currencies.

- Integrated with an external currency conversion API to provide real-time exchange rates.
- Designed a clean and intuitive UI for users to select currencies and input amounts.
- Implemented state management using React's useState and useEffect for handling and updating conversion data.

12. Password Generator

A simple yet powerful password generator tool that helps users create strong, secure passwords with customizable options.

- Built using React to provide an interactive user interface.
- Allows users to customize the length and complexity of the generated password (e.g., including uppercase letters, numbers, and special characters).
- Implemented state management using React's useState and useEffect to dynamically update the password as users adjust settings.
- Ensured the password meets security best practices by generating random and unpredictable combinations.
- Styled the application for a user-friendly experience with Tailwind CSS.

11. Tic-Tac-Toe Game

A classic and interactive two-player Tic-Tac-Toe game, offering both challenge and fun for players.

- Developed using React for the frontend to handle the dynamic nature of the game board and user interactions.
- Implemented game logic to determine win conditions, draw scenarios, and alternating turns between players.
- Managed state using React's useState and useEffect hooks for real-time game updates.
- Designed a simple, user-friendly interface with minimalistic styling to focus on the gameplay experience.
- Added a reset functionality to restart the game after each round, ensuring a seamless user experience.

13. Quiz App

An interactive quiz application allowing users to answer questions, track progress, and receive results at the end.

- Built using React for the frontend, ensuring a smooth user interface and real-time updates.
- Integrated a timer that counts down for each question, adding an element of challenge.
- Developed features to display multiple-choice questions and allow users to select answers.
- Managed user answers and scoring system, providing immediate feedback after each answer.
- Designed an engaging UI with responsive design using Tailwind CSS to ensure compatibility across devices.
- Implemented state management with React hooks (useState, useEffect) to handle current question and progress.

14. Background Color Changer

A dynamic web application that allows users to easily change the background-color of the page with a simple click.

- Built using React for a responsive and interactive frontend.
- Enables users to switch between different background colors with just a button click.
- Uses useState hook for managing state to change the background color.
- Styled with Tailwind CSS for a modern look and smooth transitions.