ABHINAV TRIPATHI

B.Tech. - Information Technology
Ph: +91-9621854341
Email: abhinavtripathi6sep@gmail.com
Allahabad, Uttar Pradesh, India - 211003

BRIEF SUMMARY

Brief Summary About Me:

I am a skilled and enthusiastic developer specializing in building robust and dynamic web applications using the MERN stack and other modern technologies. With a strong foundation in frontend development, I have developed projects like the "Crypto Tracker," a platform for tracking real-time cryptocurrency data and managing personalized watchlists and portfolios. My work demonstrates a deep understanding of integrating APIs, managing state with Redux, and creating responsive, user-friendly interfaces using React and Tailwind CSS. Additionally, I have a solid grasp of backend development with Node.js and Express, ensuring secure and efficient data handling and user authentication. I am continuously expanding my knowledge in the MERN stack, AWS, and other advanced tools, aiming to create interactive and appealing digital experiences.

KEY EXPERTISE

- Front-End Web Development
- HTML5
- CSS
- CSS Flexbox
- Javascript
- Bootstrap
- TAILWIND CSS
- AWS
- Python
- Django
- PHP
- MySQL

EDUCATION

Babu Banarasi Das Educational Group 2021 - 2025
B.Tech. - Information Technology | Percentage: **71.03 / 100**

Maharshi Patanjali Vidya Mandir, Allahabad 2016
10th | CBSE | CGPA: **7.60 / 10**

PROJECTS

Crypto Tracker

18 Jul, 2024 - 06 Sep, 2024

Key Skills: ReactJS MongoDB React Native Express.js Javascript Redux HTML5 Tailwind css

Crypto Tracker Project Overview
Project Name: Crypto Tracker
Technology Stack: MERN (MongoDB, Express.js, React, Node.js)
Purpose: To provide users with real-time cryptocurrency data, tracking prices, market trends, and portfolio management.

Objective: The primary objective of the Crypto Tracker is to offer users a platform to monitor various cryptocurrencies, including price updates, historical data, and market insights. Users can also create a personalized watchlist and portfolio to track their investments and receive alerts based on specific market conditions.

1. Introduction
Project Scope: The Crypto Tracker aims to serve as a comprehensive tool for cryptocurrency enthusiasts, investors, and traders. It allows users to view real-time market data of various cryptocurrencies, manage a personalized watchlist, and maintain a portfolio for tracking investment performance. The platform fetches data from public APIs like CoinGecko or CoinMarketCap, offering users up-to-date information on price changes, market caps, and trading volumes.

Target Audience: The Crypto Tracker is designed for crypto investors, traders, and anyone interested in tracking cryptocurrency markets. It caters to users of all experience levels, from beginners to advanced traders, providing essential tools to stay informed and make data-driven decisions.

2. Technology Stack
Frontend:

React.js: Used for building the user interface, providing a responsive and dynamic experience with modern components and hooks.
Redux/Context API: State management for handling global states such as user authentication, watchlist, and portfolio data.
Tailwind CSS/Bootstrap: For styling and creating a responsive design, ensuring the app looks good on all devices.
Backend:

Node.js: The runtime environment for executing server-side JavaScript code.
Express.js: A fast, unopinionated web framework for Node.js, used for building the RESTful API that handles requests from the frontend.
Database:

MongoDB: A NoSQL database for storing user data, watchlists, portfolios, and cached cryptocurrency data.
API Integration:

CoinGecko/CoinMarketCap API: Third-party APIs used for fetching live cryptocurrency data, including prices, market capitalization, and historical data.
Version Control:

Git & GitHub: Used for version control, collaboration, and deployment management.
Development Tools:

Visual Studio Code: The primary code editor used for developing the project.
Postman: For testing API endpoints and ensuring correct data flow between the frontend and backend.
3. Features of Crypto Tracker
1. User Authentication:

Sign Up/Sign In: Users can register using email and password. Authentication is handled using JWT (JSON Web Tokens) for secure access to personalized features.
Profile Management: Users can update their profile information and manage account settings.
2. Real-Time Cryptocurrency Data:

Live Market Data: Displays real-time prices, market caps, and volumes for hundreds of cryptocurrencies.
Top Gainers/Losers: Highlights the top-performing and worst-performing cryptocurrencies within a specified timeframe.
3. Watchlist:

Add to Watchlist: Users can add cryptocurrencies to a watchlist for quick access and monitoring.
Custom Notifications: Users can set up alerts for specific price changes or market movements.
4. Portfolio Management:

Track Investments: Allows users to add their crypto investments, including the amount bought, purchase price, and date of acquisition.
Performance Analysis: Calculates the current value of the portfolio, showing profit/loss percentages based on real-time data.
5. Historical Data and Charts:

Price Charts: Interactive charts showing price movements over various time frames (1 day, 1 week, 1 month, 1 year).
Historical Data: Provides historical price data for each cryptocurrency, enabling users to analyze trends.
6. Search and Filter:

Search Functionality: Allows users to search for specific cryptocurrencies by name or symbol.
Filters: Filter options based on market cap, volume, or price changes to help users quickly find relevant cryptocurrencies.
7. Responsive Design:

Mobile Friendly: The app is fully responsive, designed to work seamlessly on mobile devices, tablets, and desktops.
8. Dark Mode:

UI Theme Toggle: Users can switch between light and dark themes, enhancing the visual appeal and usability.
4. Detailed Implementation
1. Project Setup:

Frontend Setup: Using create-react-app or Vite for initializing the React project, setting up the basic structure with components, hooks, and state management tools like Redux or Context API.
Backend Setup: Initializing a Node.js project, setting up Express.js as the web framework, and configuring necessary middleware (body-parser, cors, etc.).
2. Database Schema Design:

User Model: Stores user information, including username, email, password (hashed), and any other profile details.
Watchlist Model: Keeps track of user-specific watchlists, including cryptocurrency IDs and alert preferences.
Portfolio Model: Records user investments, including coin details, purchase price, amount, and timestamps.
3. User Authentication:

JWT Implementation: Securely authenticates users with JSON Web Tokens, handling login, registration, and protected routes for accessing watchlists and portfolios.
Password Hashing: Using bcrypt to hash passwords before storing them in the database.
4. API Integration:

Fetching Data: Using Axios or Fetch API to retrieve data from CoinGecko or CoinMarketCap. The backend server caches this data to reduce API calls and improve performance.
Real-Time Updates: Implementing WebSockets or regular polling to keep the data up-to-date in real-time.
5. Frontend Development:

Components: Modular React components for different parts of the application, such as the Dashboard, Watchlist, Portfolio, and individual Crypto Detail pages.
Routing: Using React Router for navigating between pages, ensuring a smooth single-page application experience.
6. State Management

Book From Seniors

10 Feb, 2023 - 03 Mar, 2023

Key Skills: HTML5 CSS CSS Flexbox Django Python MySQL Javascript

Here's an updated description of your "Books from Seniors" project, where students and seniors can add any book, and students can download them in PDF format from the site for reading:

1. Introduction
Project Overview: "Books from Seniors" is a web application developed using Python and Django that serves as a digital library, enabling students and seniors to share academic resources. The platform allows users to upload any academic book in PDF format, making it accessible to the entire student community. Students can browse and download these books directly from the site, providing a centralized, user-friendly, and resourceful environment for accessing study materials.

Purpose and Goals: The primary goal of the "Books from Seniors" website is to create an accessible and sustainable resource-sharing platform that helps students obtain academic books in PDF format. By digitizing books and making them available online, the platform reduces costs and promotes easy access to educational materials. It also encourages collaboration among students by sharing resources that are otherwise difficult to obtain.

2. Technology Stack
Backend:

Python: The backend logic is built with Python, leveraging its robust and straightforward nature.
Django: Django is used as the web framework, providing features like ORM, built-in authentication, and a powerful admin panel.
Frontend:

HTML5, CSS3, JavaScript: These standard web technologies are used for creating the layout, design, and interactivity of the website.
Bootstrap/Tailwind CSS: A CSS framework to ensure the site is responsive and has a modern design.
Database:

SQLite/PostgreSQL/MySQL: Depending on the deployment needs, SQLite is used in development, with PostgreSQL or MySQL as options for production environments.
Additional Tools:

Git & GitHub: For version control and project collaboration.
VS Code/PyCharm: Used as the primary development environments for coding and debugging.
3. Features of the Books from Seniors Website
1. User Registration and Authentication:

Sign Up/Sign In: Users can create accounts using their email, set up profiles, and log in to access the platform's features.
Password Reset: Secure password reset functionality using email verification.
2. Home Page:

A welcoming landing page displaying featured books, new uploads, and an overview of how to use the platform.
3. Book Listings and Uploads:

Browse Books: Users can explore available books categorized by subject, academic year, or department.
Search and Filters: Advanced search functionality with filters for subject, academic year, and file type.
Upload Books: Students and seniors can upload books in PDF format, providing necessary details such as title, author, and category.
4. Book Details and Download:

Detailed View: Each book has a dedicated page with its details, including a download button for students to access the PDF file.
PDF Download: Students can download the books directly in PDF format for offline reading.
5. Book Management:

Manage Uploads: Users can view and manage their uploaded books, including options to edit or delete their listings.
6. Admin Panel:

Content Management: Administrators can manage user-uploaded books, moderate content, and handle any disputes or issues that arise.
Reports and Analytics: Generate reports on book uploads, downloads, and user activity for monitoring the platform's performance.

College Website

04 Aug, 2022 - 10 Sep, 2022

Key Skills: HTML5 CSS CSS Flexbox Python Django Bootstrap MySQL Javascript

The "College Website" project is a comprehensive web application developed using Python and the Django framework. This project aims to provide a dynamic and user-friendly platform for students, faculty, and administrators to access and manage various academic and administrative information. The website includes features such as course management, faculty information, event listings, student registrations, and more, offering a centralized platform for a college or university.

Purpose and Goals: The primary goal of the College Website is to streamline the information flow within the institution. It serves as a portal for disseminating information about academic programs, faculty details, and upcoming events. Additionally, it simplifies the process of student registrations, announcements, and course updates, thereby enhancing the overall efficiency and communication within the college.

2. Technology Stack
Backend:

Python: The primary programming language used for the backend. It provides simplicity and ease of use for rapid development.
Django: A high-level Python web framework that encourages rapid development and clean, pragmatic design. It comes with a built-in admin panel, ORM, and user authentication system, which significantly reduces development time.
Frontend:

HTML5, CSS3, JavaScript: The core technologies for the frontend, providing structure, styling, and interactivity.
Bootstrap/Tailwind CSS: A CSS framework for building responsive, mobile-first projects on the web.
Database:

SQLite/MySQL/PostgreSQL: Django's ORM allows easy integration with various databases. The choice depends on the scale and requirements of the project.
Other Tools:

Git & GitHub: For version control and collaboration.
VS Code/PyCharm: For code development and debugging.
3. Features of the College Website
1. User Authentication and Authorization:

Login and Registration: Users can register and log in to access personalized features. There are different user roles such as students, faculty, and administrators, each with specific access rights.
Password Recovery: A secure password recovery mechanism using email verification.
2. Home Page:

A dynamic and informative landing page that provides a snapshot of the college's key highlights, such as news, announcements, featured courses, and upcoming events.
3. Course Management:

Course Listings: Students can browse through available courses, view detailed descriptions, faculty information, and schedules.
Enrollment System: An intuitive system for students to enroll in courses, with real-time updates on available slots and prerequisites.
4. Faculty Module:

Faculty Directory: A searchable directory of faculty members, including their profiles, qualifications, contact information, and courses they teach.
Office Hours: Faculty can set and manage their office hours, allowing students to schedule appointments.
5. Student Dashboard:

Personalized Dashboard: Displays enrolled courses, grades, announcements, and upcoming events tailored to the student's profile.
Profile Management: Allows students to update personal information, upload profile pictures, and manage their academic records.
6. Event Management:

Event Calendar: A comprehensive calendar that lists all upcoming college events, seminars, and workshops. Users can filter events by category or department.
Event Registration: Students can register for events, and faculty can manage event participation.8. Admin Panel:

Content Management: Administrators can manage courses, faculty, students, and events directly from a centralized dashboard.
Reports and Analytics: Generate reports on student enrollment, course popularity, and event attendance to aid in decision-making.

PERSONAL INTERESTS / HOBBIES

- FREE HAND SKETCHING
- TATTO MAKING
- LISTENING MUSIC

IMs

- WhatsApp - 9452100940

PERSONAL DETAILS

Gender: Male

Marital Status: Single

Current Address: 90 E Malak Raj New Baharana Allahabad, Allahabad, Uttar Pradesh, India - 211003

Email: abhinavtripathi6sep@gmail.com

Date of Birth: 06 Sep, 2000

Known Languages: Hindi,English

Phone Number: +91-9621854341