01)Name : Campus Event Management System

02)Description : The Campus Event Management System is a web-based application designed to manage college events efficiently. It provides separate portals for staff and students, where staff can create and manage events, and students can view events, register, give feedback, and track attendance. The system simplifies the entire event lifecycle in a single platform.

03)Problem Solved : In many colleges, event management is handled manually using WhatsApp groups, notices, or spreadsheets. This leads to poor communication, difficulty in tracking registrations, attendance errors, and loss of feedback data.
This project solves these problems by providing a centralized digital platform that automates event creation, student registration, attendance tracking, and feedback collection in a structured and reliable way.

04)Contribution : I was responsible for designing and developing both the Student Portal and Staff Portal interfaces.
I implemented features such as event listing, event registration, attendance tracking, feedback submission, and dashboard statistics. I also handled frontend–backend integration, API handling, deployment on Vercel, and debugging real production issues like CORS and environment configuration.

05)Tools Used : a)Frontend : Next.js, React, TypeScript, Tailwind CSS
b)Backend : Python (Flask) REST APIs
c)Database : MySQL
d)Deployment : Vercel (Frontend),backend(railway.app)
e)Other Tools : GitHub, REST APIs, Service Workers (PWA)

01)Project Name : Blocktix–Blockchain-Based Event Ticketing System

02)Description : Blocktix is a decentralized event ticketing platform designed to make ticket creation, distribution, and verification more secure and transparent. The system uses blockchain concepts to prevent ticket duplication and fraud while providing a simple interface for users to view events and manage tickets. It demonstrates how modern web applications can integrate decentralized technologies for real-world use cases.

03)Problem Solved : Traditional event ticketing systems often face issues such as fake tickets, duplicate entries, lack of transparency, and centralized control. Blocktix addresses these problems by introducing a blockchain-based approach where ticket data is securely stored and verifiable. This improves trust, reduces fraud, and ensures authenticity of event tickets.

04)Contribution : I worked on developing both the frontend and backend of the application. My responsibilities included building the user interface, implementing ticket creation and validation logic, integrating blockchain-related concepts, and handling API communication between the frontend and backend. I also managed version control and deployment of the project.

05)Tools Used : a)Frontend : React, Vite, TypeScript, Tailwind CSS
b)Backend : Node.js, Express.js
c)Blockchain : Ethereum / Smart Contract concepts
d)Other Tools : GitHub, REST APIs
e)Deployment : Vercel (Frontend), Railway (Backend)

01)Project Name : Currency Converter Web Application

02)Description : The Currency Converter Web Application is a simple and user-friendly web tool that allows users to convert amounts between different currencies using real-time exchange rates. The application provides an interactive interface with country flags and dropdowns to select currencies, making it easy for users to understand and use.

03)Problem Solved : Users often need to quickly check exchange rates when dealing with international transactions, travel planning, or online purchases. Searching for rates repeatedly can be inefficient. This project solves the problem by offering instant and accurate currency conversion in one place using a real-time exchange rate API.

04)Contribution : I independently designed and developed the complete application. My contribution included building the UI using HTML and CSS, implementing the conversion logic using JavaScript, integrating a public exchange rate API, and handling user interactions such as currency selection and input validation. I also deployed the project and ensured it works reliably as a static web application.

05)Tools Used : a)Frontend : HTML, CSS, JavaScript
b)API : Open Exchange Rate API
c)Deployment : Vercel / GitHub Pages
d)Other Tools : GitHub, REST APIs