

NEW PROJECT SET - WEB SOFTWARE DEVELOPMENT (MERN STACK) —STARTUP REALITY FOCUS

Real Startup Use Cases – Hands-on Product Building

Project 1 - "GigConnect": A Hyperlocal Freelance Marketplace

Problem Statement: *To create a streamlined platform that connects local communities with skilled freelancers, enabling clients to easily find and hire local talent for specific services while providing freelancers with a platform to showcase their skills and find nearby job opportunities.*

Use Case: Develop a mobile-responsive web application where users can register as either a client or a freelancer. Clients can post job listings (gigs), search for freelancers based on skill and location, and manage payments securely. Freelancers can create detailed profiles, browse and apply for local gigs, and communicate with clients directly through the platform.

Key Modules:

- **Dual-Role User Authentication:** Secure registration and login for both "Client" and "Freelancer" roles using JWT.
 - **Freelancer Profile Management:** Detailed profiles with skills, portfolio, service rates, and user reviews.
 - **Gig Posting & Management:** Clients can create, edit, and manage job postings with clear requirements.
 - **Advanced Search & Filtering:** Hyperlocal search functionality with filters for skills, location, price range, and ratings.
 - **Real-time Messaging System:** Integrated chat for seamless communication between clients and freelancers.
 - **Review & Rating System:** A two-way feedback system to build trust and reputation.
 - **Secure Payment Integration:** Integration with Razorpay/Stripe for secure transactions and milestone payments.
-

Week-wise Development Plan:

Week	Backend (Node.js + Express)	Frontend (React.js)
Week 1	Setup JWT Authentication for dual roles (Client/Freelancer), build APIs for user profiles and gig postings (CRUD).	Create React setup, build UI for login/registration, freelancer profile pages, and the gig posting form.
Week 2	Develop backend logic for advanced search, filtering APIs, and set up the real-time chat with Socket.IO.	Implement the main gig feed with search and filter functionality, and build the real-time messaging interface.
Mid Project Review	Core APIs for users, gigs, and search are functional. Real-time chat is initiated.	Users can register, create profiles, post gigs, and search for freelancers. Basic chat UI is in place.
Week 3	Implement the review and rating system APIs. Develop APIs for the secure payment workflow.	Build the UI for submitting and displaying reviews. Integrate the payment gateway on the frontend.
Week 4	Create Admin dashboard APIs for managing users and gigs. Conduct final API testing and cleanup.	Develop the Admin panel UI. Perform final UI polishing and ensure the application is fully responsive.
Final Project Review	All modules including payments, reviews, and chat are fully functional and secure.	A complete, responsive frontend with a seamless user experience for both clients and freelancers.

Project 2 - "SyncSpace": A Real-time Collaborative Workspace & Project Hub

Problem Statement: *Develop an all-in-one collaborative platform that allows teams to manage projects, share documents, and communicate in real-time, eliminating the need for multiple disconnected tools and improving overall team productivity.*

Use Case: Build a sophisticated tool where teams can create dedicated workspaces for different projects. Within each workspace, users can utilize a Kanban board for task management, a real-time collaborative document editor, and an integrated chat, ensuring all project-related activities are centralized and synchronized across all team members.

Key Modules:

- **User & Team Management:** Secure user authentication and role-based access control (Admin, Member) within teams.
- **Workspace & Project Creation:** Ability for users to create and manage distinct project workspaces.
- **Dynamic Kanban Boards:** Interactive, drag-and-drop task boards for visual project management.
- **Real-time Document Editor:** A collaborative editor (similar to Google Docs) allowing multiple users to edit a document simultaneously.
- **Integrated Chat & Commenting:** Real-time chat channels for each workspace and commenting threads on individual tasks.
- **File Sharing & Version Control:** A centralized repository for project files with basic version history.
- **Notifications System:** Real-time alerts for task assignments, mentions, and project updates.

Week-wise Development Plan:

Week	Backend (Node.js + Express)	Frontend (React.js)
Week 1	Implement JWT authentication, APIs for user/team management, and CRUD APIs for workspaces and Kanban boards.	Set up React project, build UI for authentication, user dashboard, and workspace creation.
Week 2	Develop real-time APIs using Socket.IO for Kanban boards and the collaborative document editor.	Build the interactive Kanban board UI with drag-and-drop functionality and the basic real-time text editor.
Mid Project Review	Core APIs for user management and project setup are complete. Real-time collaboration initiated.	The collaborative Kanban board is functional, and users can manage tasks in real-time.
Week 3	Build APIs for the integrated chat, task commenting system, and file upload/versioning logic.	Develop the UI for the real-time chat module, task detail view with comments, and file management system.
Week 4	Implement the notification system APIs. Finalize API testing, security hardening, and cleanup.	Integrate the notifications UI. Focus on UI polishing, state management refinement, and responsiveness.

Week	Backend (Node.js + Express)	Frontend (React.js)
Final Project Review	All modules, including real-time collaboration on boards and docs, are fully functional and integrated.	A fully responsive and interactive frontend for the complete collaborative workspace.

Project 3 - "TasteTrail": An Interactive Recipe Discovery & Meal Planning App

Problem Statement: *Design a personalized recipe and meal planning application that helps users discover new dishes based on their dietary preferences, efficiently plan their weekly meals, and automatically generate shopping lists to simplify their cooking experience.*

Use Case: Develop an engaging platform where users can create a profile detailing their dietary needs (e.g., vegan, gluten-free), allergies, and taste preferences. The app will provide personalized recipe recommendations, allow users to build a weekly meal plan via a drag-and-drop calendar, and auto-generate a categorized shopping list based on the selected meals.

Key Modules:

- **User Profile & Dietary Preferences:** User registration with detailed settings for diet, allergies, and cuisines.
- **Smart Recipe Discovery Engine:** A database of recipes with advanced filtering (by ingredient, prep time, diet, etc.).
- **Interactive Meal Planner:** A weekly calendar interface where users can drag and drop recipes to plan meals.
- **Automated Shopping List Generator:** Automatically compiles an organized shopping list from the user's meal plan.
- **Recipe Saving & Collections:** Users can save favorite recipes and organize them into personal collections (e.g., "Quick Dinners").
- **User Reviews & Photo Sharing:** Community feature for users to rate recipes and upload photos of their creations.
- **Admin Panel for Recipe Management:** An interface for administrators to add, edit, and categorize recipes.

Week-wise Development Plan:

Week	Backend (Node.js + Express)	Frontend (React.js)
Week 1	Set up user authentication, build APIs for user profiles and dietary preferences, and implement CRUD APIs for recipes.	Create React setup, build UI for user registration/login, profile setup, and recipe Browse/viewing pages.
Week 2	Develop APIs for the meal planner (saving/updating weekly plans) and the recipe saving/collections feature.	Build the recipe discovery page with search and filtering. Implement the "save recipe" functionality.
Mid Project Review	Core APIs for users and recipes are functional. Users can be authenticated and view recipe data.	Users can register, set preferences, and browse/search for recipes. The main discovery UI is functional.
Week 3	Implement backend logic for the automated shopping list generator. Develop APIs for user reviews and ratings.	Build the drag-and-drop meal planner calendar UI. Develop the UI for submitting and viewing recipe reviews.
Week 4	Build APIs for the Admin panel (recipe management). Conduct final testing and API optimization.	Develop the UI for the generated shopping list and the Admin dashboard. Finalize UI polish and responsiveness.
Final Project Review	All modules, including the meal planner and shopping list generator, are fully functional.	A complete, responsive frontend allowing users to discover recipes, plan meals, and generate shopping lists.