

# **CLIENTSYNC – FREELANCER CRM (REACT + JSON SERVER)**

A Project Report

*Submitted by*

STUDENT'S

Syed Arsh - AP23110010587

Surya Teja Annam – AP23110010525

Hrushikesh Bhaskar Gopale – AP23110010129

Hitesh Karthikeya Adusumilli – AP23110010480

*In partial fulfilment for the requirements of the Final Project*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**SRM**  
**UNIVERSITY AP**

—————  
**Andhra Pradesh**

## INTRODUCTION

Welcome to ClientSync, an elegant and modern CRM application designed specifically for freelancers and small businesses. Built using the power of **React.js**, **Tailwind CSS**, and **JSON Server**, ClientSync simplifies client management, project tracking, and financial invoicing — all under one streamlined dashboard.

ClientSync combines simplicity with functionality, giving freelancers a centralized platform to organize their workflow effortlessly. Whether you're managing multiple clients, handling multiple projects simultaneously, or keeping track of payments and invoices, ClientSync ensures everything is accessible, fast, and beautifully presented.

Crafted using React's component-driven architecture, the UI is responsive, engaging, and easy to use. Every section—from client management to revenue insights—is designed with clarity and efficiency in mind, ensuring a smooth experience across all devices.

ClientSync redefines how freelancers operate daily by turning complex workflows into an enjoyable and productive experience.

## SCENARIO-BASED INTRO

Imagine you're a freelance developer or designer managing four clients at once. Each client has ongoing projects, and each project has invoices with different due dates.

Normally, keeping track of all this manually becomes stressful. But with **ClientSync**, your workday becomes organized and stress-free.

You open the app and instantly see:

- Total clients
- Total ongoing projects
- Total invoices
- Revenue paid vs. pending

You click on the *Clients* section to update a client's email, then move to the Projects tab to change a project's status from "In Progress" to "Completed".

Finally, in the Invoices section, you mark an invoice as "Paid", and instantly the Dashboard updates your revenue insights.

With ClientSync, your daily freelance workflow becomes automated, organized, and visually clean.

## **TARGET AUDIENCE**

ClientSync is designed for:

### **1. Freelancers**

Developers, designers, writers, marketers, and other freelancers who need a simple CRM to manage clients, projects, and invoices.

### **2. Small Agencies**

Small teams or startups who manage multiple clients and projects.

### **3. Independent Consultants**

Professionals who track billable work, payments, and clients.

# PROJECT GOALS AND OBJECTIVES

The main goal of ClientSync is to provide freelancers with a lightweight yet powerful CRM solution.

## Objectives:

### 1. User-Friendly Interface

Create a clean, modern UI that helps users manage clients, projects, and invoices effortlessly.

### 2. Organized Data Management

Provide structured data handling using JSON Server for CRUD operations.

### 3. Insightful Dashboard

Offer real-time analytics like:

- Total clients
- Revenue statistics
- Recent activity
- Project status distribution

### 4. Modern Tech Stack

Use React.js, Tailwind CSS, Axios, React Router, and JSON Server to build a fast, responsive, and interactive experience.

## KEY FEATURES

- **Dashboard Overview**

A visually appealing dashboard showing summary statistics, charts, and recent activity.

- **Client Management**

Add, edit, delete, and view client details (email, phone, company, etc.).

- **Project Management**

Track project status, amount, associated clients, and deadlines.

- **Invoice Management**

Generate invoices, assign them to clients and projects, mark as paid or pending.

- **Real-Time Statistics**

Revenue insights, project progress tracking, and recently added data.

- **Clean Navigation**

Navbar with smooth routing between Dashboard, Clients, Projects, and Invoices.

- **JSON Server Backend**

Stores and manages all CRUD operations easily.

- **Responsive Design**

Fully responsive UI made using Tailwind CSS.

## **PRE-REQUISITES**

### **Node.js and npm**

To run JavaScript and React applications locally.

Download: <https://nodejs.org/en/download/>

### **React.js**

Modern library for building user interfaces.

Commands:

```
npm create vite@latest
cd clientsync
npm install
npm run dev
```

### **Tailwind CSS**

Utility-first CSS framework for styling.

### **JSON Server**

Used as a lightweight backend for storing clients, projects, and invoices.

### **Axios**

For making API calls to JSON Server.

### **React Router**

For handling navigation across multiple pages.

### **Code Editor (VS Code)**

Recommended for writing code.

## Version Control

Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

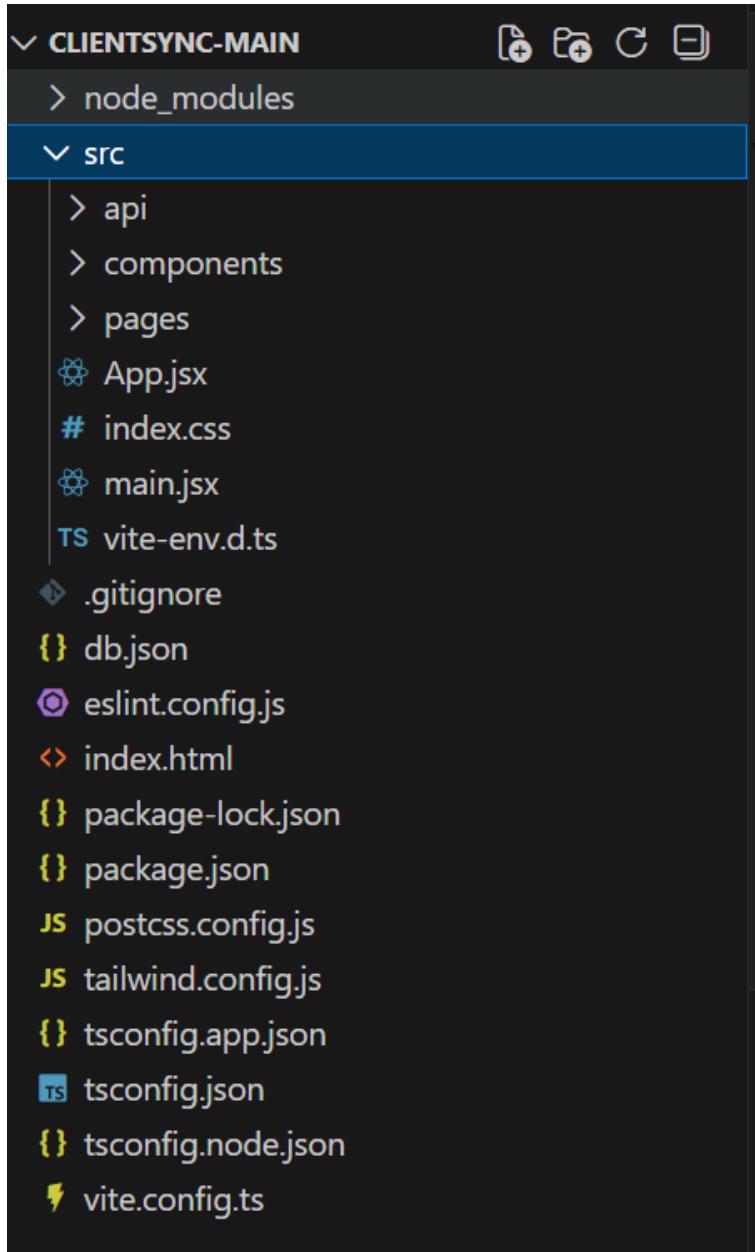
- **Git:** Download and installation instructions can be found at: <https://git-scm.com/downloads>
- **Development Environment:** Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

Visual Studio Code: Download from <https://code.visualstudio.com/download>

Sublime Text: Download from <https://www.sublimetext.com/download>

WebStorm: Download from <https://www.jetbrains.com/webstorm/download>

## PROJECT STRUCTURE



The **App.jsx** handles routing.

The **pages** folder contains all CRUD views.

The **api.js** file manages all API requests.

# PROJECT FLOW

## 1. Project Setup

- Created the project using Vite
- Installed React Router, Axios, JSON Server, Tailwind CSS
- Set up API base URL and CRUD routes

## 2. UI Development

- Built the Dashboard
- Created pages for Clients, Projects, Invoices
- Added forms for Add/Edit functionality

## 3. API Integration

- Connected CRUD pages with JSON Server
- Used Axios for GET, POST, PUT, DELETE
- Synced Dashboard insights with live data

## 4. Styling

- Implemented Tailwind CSS utilities
- Added custom gradients and themes
- Designed responsive layout

## 5. Project Execution

- Run backend:  
`npm run server`
- Run frontend:  
`npm run dev`
- The app opens at:  
`http://localhost:5173`

The app launches with the Dashboard as the home page.

```
PS C:\Users\ASUS\OneDrive\Desktop\Sem - 5\FS\ClientSync-main> npm run dev
> clientsync@1.0.0 dev
> concurrently -n "JSON-SERVER,VITE" -c "blue,green" "npm run server" "vite"

[JSON-SERVER]
[JSON-SERVER] > clientsync@1.0.0 server
[JSON-SERVER] > json-server --watch db.json --port 5000
[JSON-SERVER]
[JSON-SERVER] --watch/-w can be omitted, JSON Server 1+ watches for file changes by default
[JSON-SERVER] JSON Server started on PORT :5000
```

```
[JSON-SERVER] Press CTRL-C to stop
[JSON-SERVER] Watching db.json...
[JSON-SERVER]
[JSON-SERVER] 💕( ˘◡˘ )
[JSON-SERVER]
[JSON-SERVER] Index:
[JSON-SERVER] http://localhost:5000/
[JSON-SERVER]
[JSON-SERVER] Static files:
[JSON-SERVER] Serving ./public directory if it exists
```

```
PS C:\Users\ASUS\OneDrive\Desktop\Sem - 5\FS\ClientSync-main> npm run dev
[JSON-SERVER] Endpoints:
[JSON-SERVER] http://localhost:5000/clients
[JSON-SERVER] http://localhost:5000/projects
[JSON-SERVER] http://localhost:5000/invoices
[VITE]
[VITE]   VITE v5.4.8 ready in 2457 ms
[VITE]
[VITE]   → Local: http://localhost:5173/
[VITE]   → Network: use --host to expose
[VITE] Browserslist: caniuse-lite is outdated. Please run:
```

# **SYSTEM REQUIREMENTS**

## **Hardware Requirements**

- Minimum 4 GB RAM
- Any modern processor (Intel i3 or above)
- 500 MB of free disk space
- Monitor with minimum 1366×768 resolution

## **Software Requirements**

- Windows / macOS / Linux
- Node.js (Latest LTS version)
- Visual Studio Code
- Google Chrome or Firefox browser
- JSON Server
- React dependencies installed

# TECHNOLOGY STACK

## Frontend – React.js

- Component-based UI
- Fast rendering
- Virtual DOM
- Smooth navigation using React Router

## Styling – Tailwind CSS

- Utility-first CSS
- Responsive by default
- Modern gradients, shadows, spacing utilities

## Backend – JSON Server

- Acts as a mock REST API
- Stores data inside db.json
- Supports GET, POST, PUT, DELETE

## HTTP Client – Axios

- For all API communication
- Handles requests and responses cleanly

## Build Tool – Vite

- Very fast development server
- Bundles optimized production build

# MODULE DESCRIPTION

## 1. Dashboard Module

- Displays total clients, projects, invoices
- Shows revenue insights
- Shows recent activity
- Provides quick action shortcuts

## 2. Clients Module

- Add new clients
- Edit client details
- Delete clients
- View full list of clients

## 3. Projects Module

- Create new projects
- Assign multiple clients
- Change project status
- Track project amounts

## 4. Invoices Module

- Generate invoices linked to clients and projects
- Mark invoices as paid or pending
- View financial details

## 5. API Module

- Handles Axios requests
- Connects frontend with JSON Server
- Reusable functions for CRUD

## **6. Navigation Module**

- Navbar for smooth routing
- Uses React Router DOM
- No page reload while navigating

## **PROJECT FLOW& CODE DEMO LINKS**

**Project Flow Demo:**

[https://drive.google.com/file/d/17sSAnwemQ\\_XgFLNWN41db2w4rxb7ZCAg/view?usp=sharing](https://drive.google.com/file/d/17sSAnwemQ_XgFLNWN41db2w4rxb7ZCAg/view?usp=sharing)

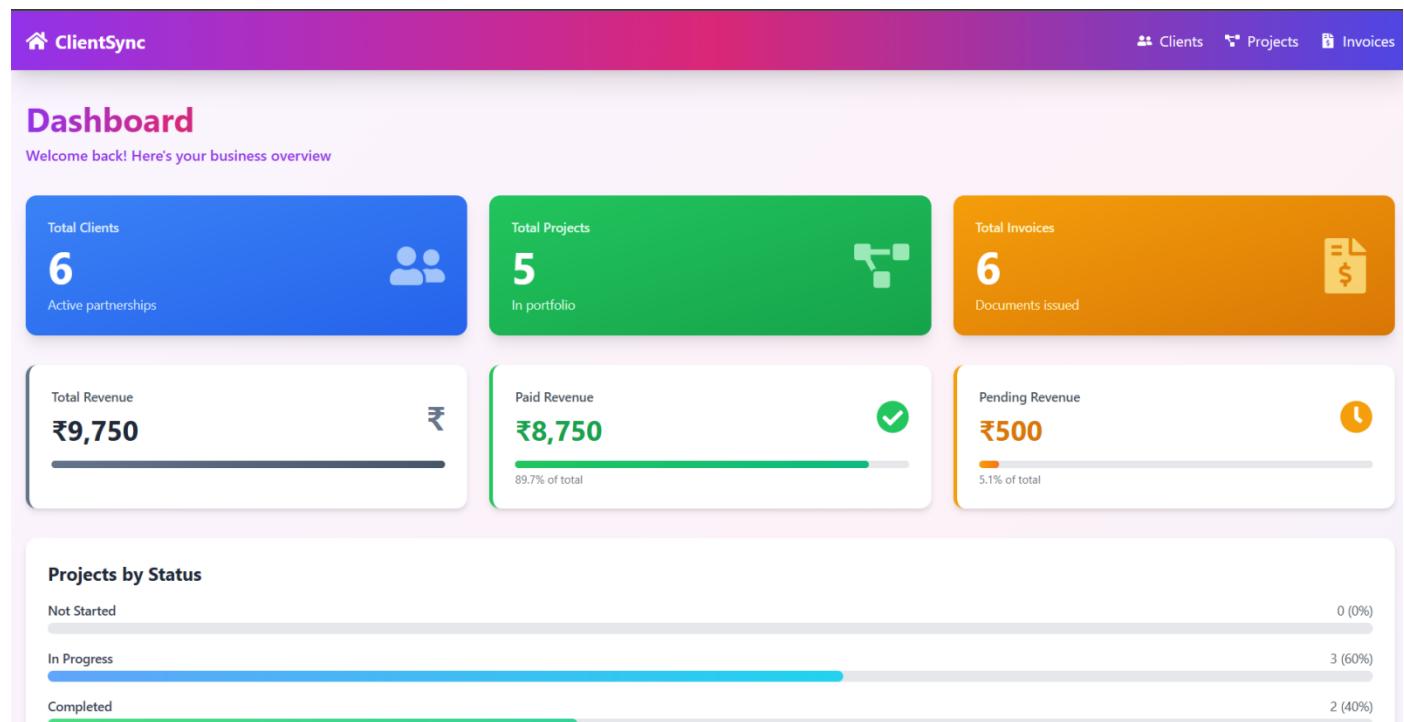
**Project Code Demo:**

[https://drive.google.com/file/d/185M\\_c223JyRU2-Pgu1FYkzQuh-lI49Kr/view?usp=sharing](https://drive.google.com/file/d/185M_c223JyRU2-Pgu1FYkzQuh-lI49Kr/view?usp=sharing)

# USER INTERFACE SCREENS

## Dashboard Screen

Shows all statistics, charts, and activity.



**Recent Activity**

- Recent Clients**
  - Asha Web Studio  
contact@ashastudio.com  
Asha Web Studio
  - arsh  
arsh@gmail.com  
srm
  - Sairam  
sairam@gmail.com  
SRM[View all clients →](#)
- Recent Projects**
  - E-commerce Website Redesign  
completed ₹5,000
  - SRM WORK  
in progress ₹1,000
  - SRM Work  
in progress ₹1,000[View all projects →](#)
- Recent Invoices**
  - INV-2025-001  
paid ₹2,500
  - INV-2025-522  
cancelled ₹500
  - INV-2025-111  
pending ₹500[View all invoices →](#)

**Quick Actions**

- [Add New Client](#)
- [Create Project](#)
- [Generate Invoice](#)

**ClientSync**

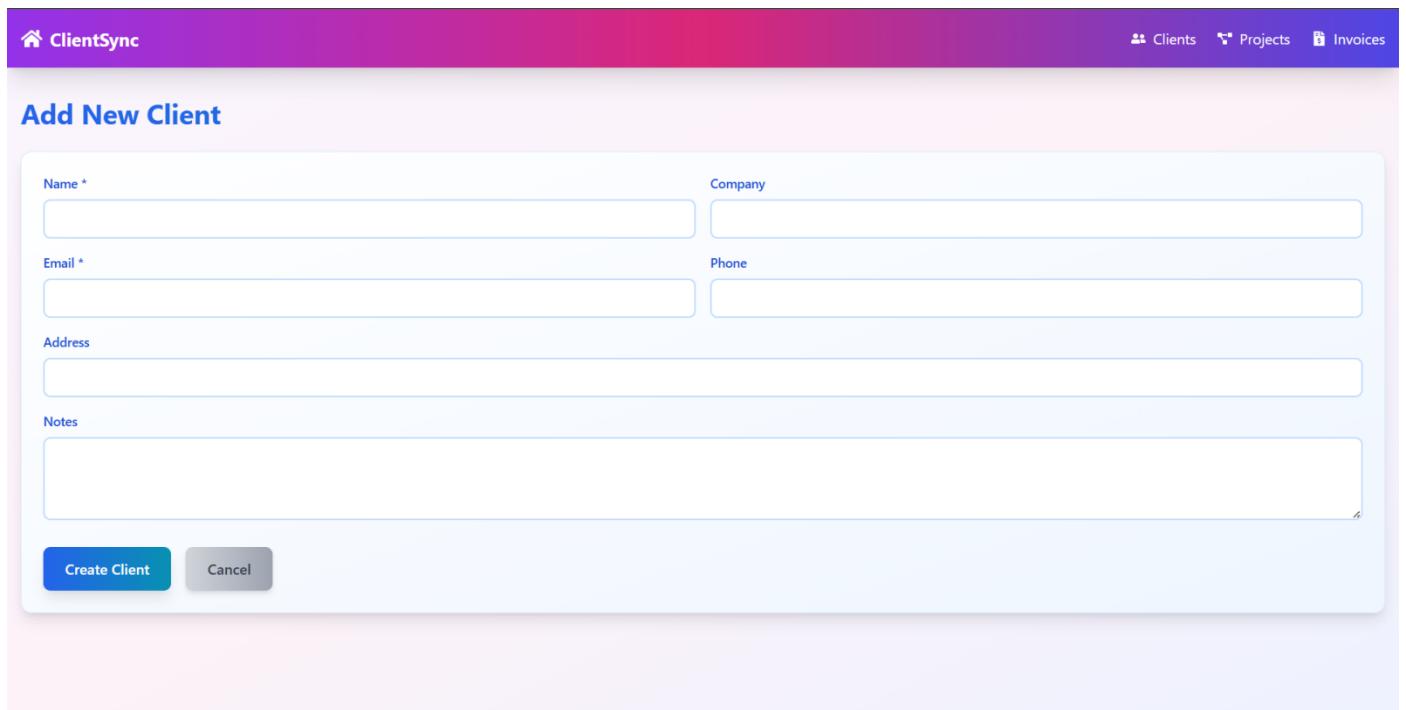
**Clients**

Name	Company	Email	Phone	Actions
Asha Web Studio	Asha Web Studio	contact@ashastudio.com	9876543210	
arsh	srm	arsh@gmail.com	1234567890	
Sairam	SRM	sairam@gmail.com	-	
Surya	-	surya@gmail.com	-	
Hitesh	-	hitesh@gamil.com	-	
rebgrsfbd	-	fbafbrw@gmail.com	-	

## Clients Screen

Table/list of all clients with buttons to edit/delete.

Clients				
<input placeholder="Search clients by name, email, or company..." type="text"/> <span style="float: right;"><a href="#">+ Add Client</a></span>				
Name	Company	Email	Phone	Actions
Asha Web Studio	Asha Web Studio	contact@ashastudio.com	9876543210	
arsh	srm	arsh@gmail.com	1234567890	
Sairam	SRM	sairam@gmail.com	-	
Surya	-	surya@gmail.com	-	
Hitesh	-	hitesh@gamil.com	-	
rebgrsfbd	-	fbafbrw@gmail.com	-	



**Add New Client**

Name \*

Company

Email \*

Phone

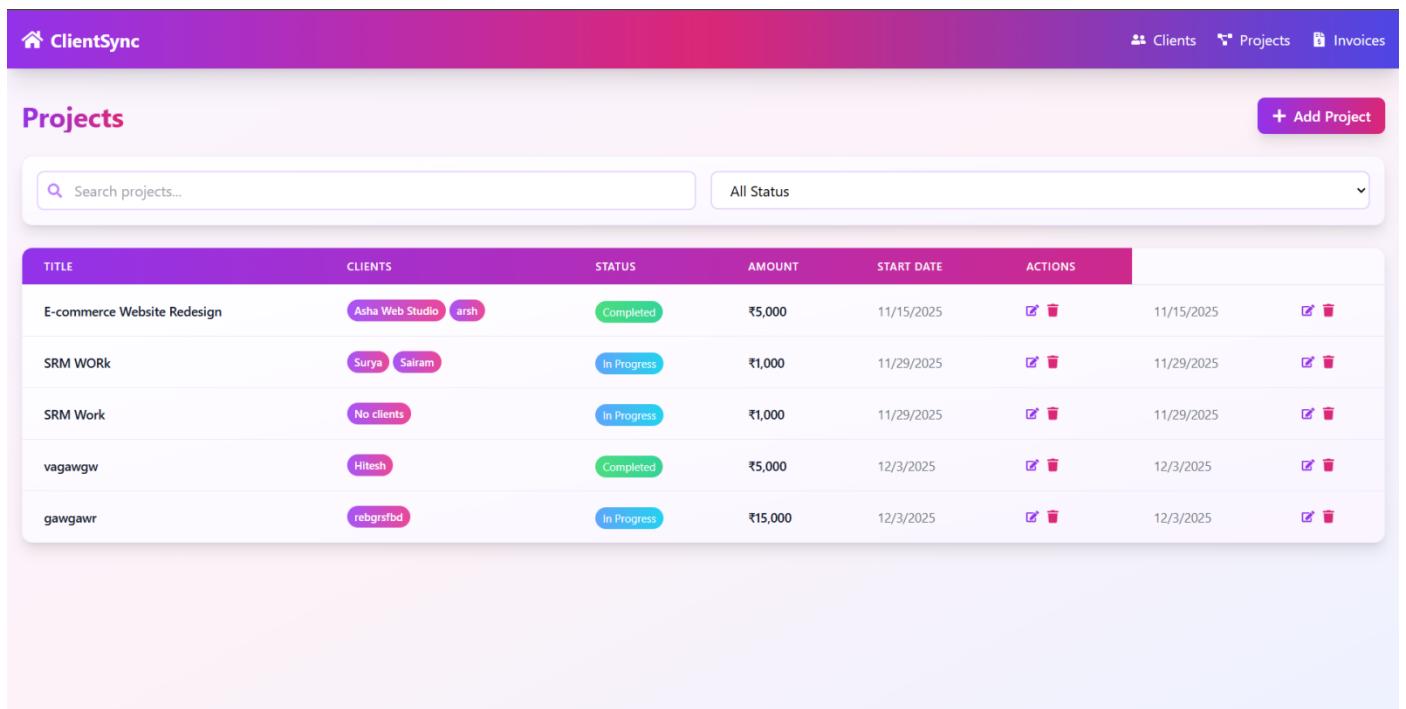
Address

Notes

**Create Client** **Cancel**

## Projects Screen

List of projects, with status badges.



**Projects** **+ Add Project**

Search projects...  All Status

Title	Clients	Status	Amount	Start Date	Actions
E-commerce Website Redesign	Asha Web Studio arsh	Completed	₹5,000	11/15/2025	
SRM WORK	Surya Salam	In Progress	₹1,000	11/29/2025	
SRM Work	No clients	In Progress	₹1,000	11/29/2025	
vagawgw	Hitesh	Completed	₹5,000	12/3/2025	
gawgawr	rebgarsfbd	In Progress	₹15,000	12/3/2025	

**Add New Project**

Clients \* (Select one or more)

<input type="checkbox"/> Asha Web Studio	<input type="checkbox"/> arsh	<input type="checkbox"/> Sairam
<input type="checkbox"/> Surya	<input type="checkbox"/> Hitesh	<input type="checkbox"/> rebgrsfbd

Title \*

Description

Status \*

Not Started

Amount \*

Start Date \*

 mm/dd/yyyy
 

End Date \*

 mm/dd/yyyy
 

**Create Project**

**Cancel**

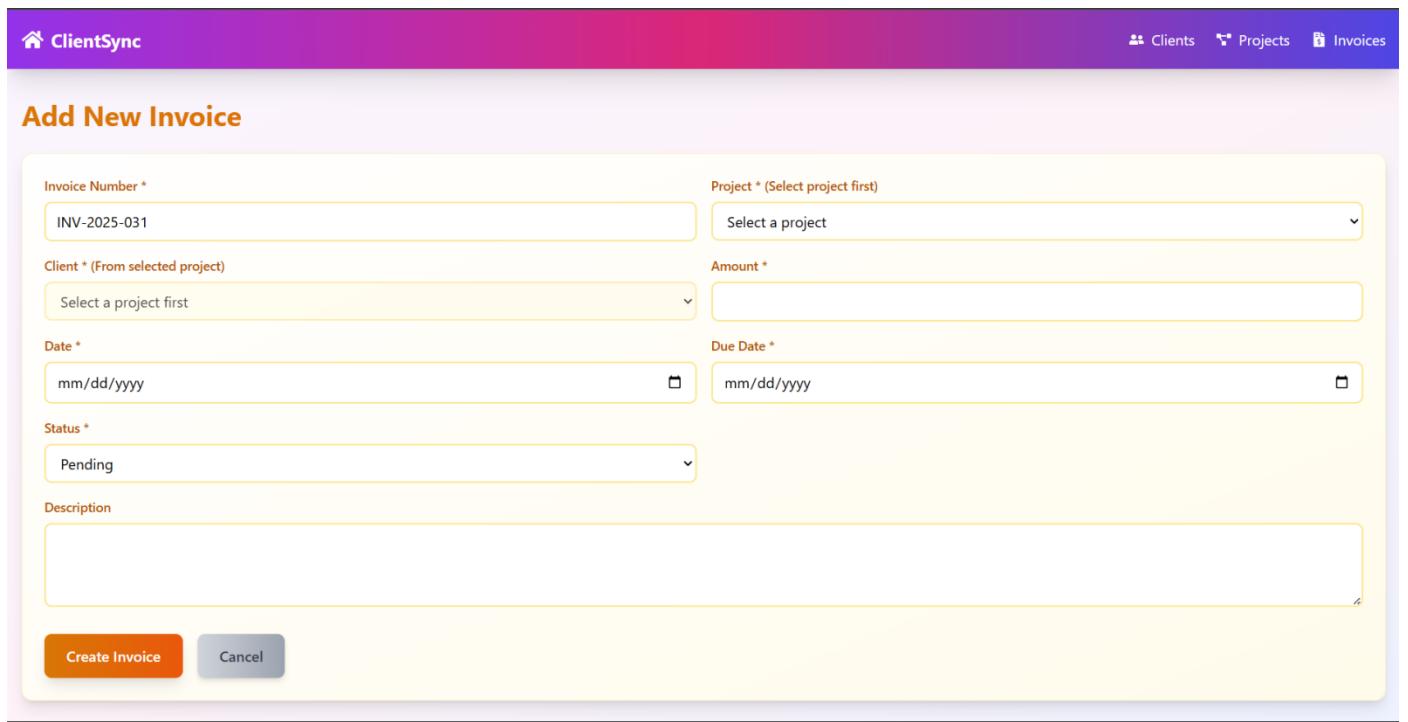
## Invoices Screen

Invoice list with color-coded statuses.

**Invoices**

**+ Add Invoice**

INVOICE #	PROJECT	CLIENT	AMOUNT	DATE	DUUE DATE	STATUS	ACTIONS	EXPORT
INV-2025-001	E-commerce Website Redesign	Asha Web Studio	₹2,500	11/20/2025	12/20/2025	Paid		
INV-2025-522	SRM WORK	Sairam	₹500	11/29/2025	12/31/2025	Cancelled		
INV-2025-111	SRM WORK	Sairam	₹500	11/29/2025	11/29/2025	Pending		
INV-2025-113	SRM WORK	Surya	₹500	11/29/2025	12/31/2025	Paid		
INV-2025-360	SRM WORK	Surya	₹750	12/3/2025	12/3/2025	Paid		
INV-2025-127	vagawgw	Hitesh	₹5,000	12/3/2025	12/3/2025	Paid		

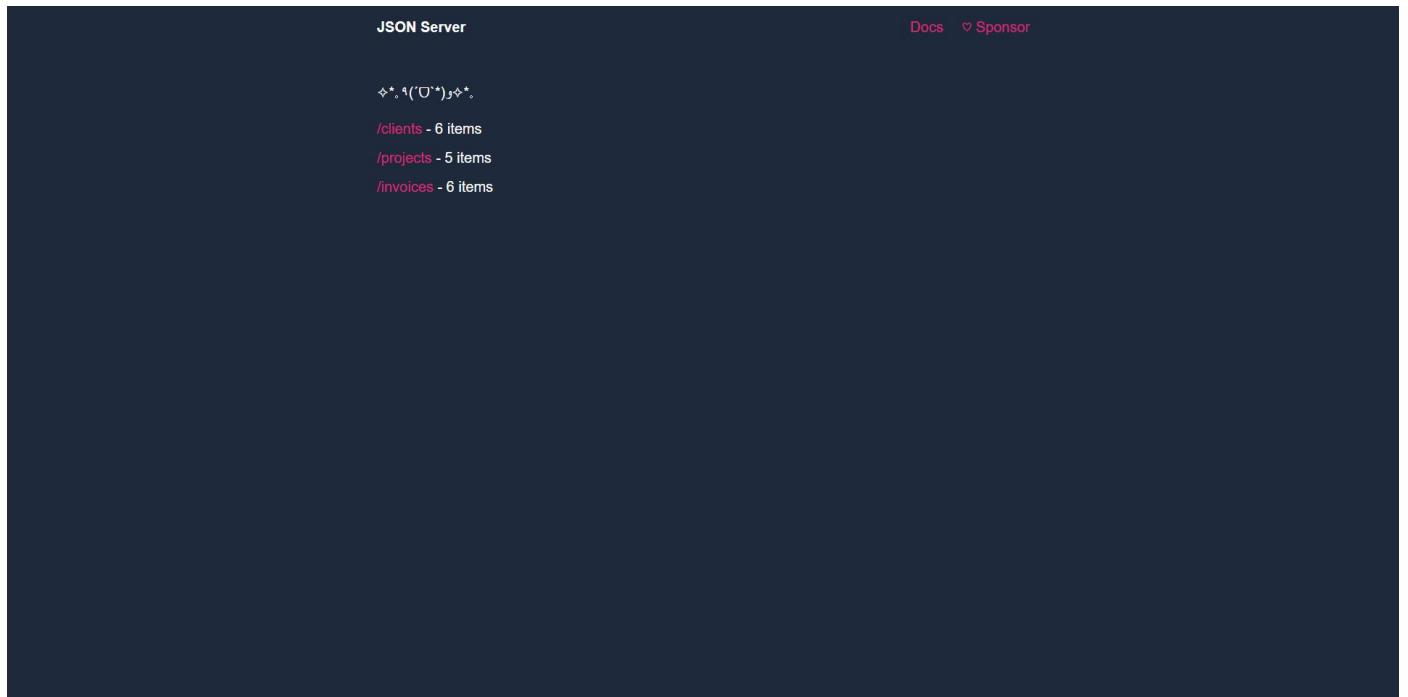


The screenshot shows the 'Add New Invoice' form in ClientSync. The form has a purple header bar with the ClientSync logo and navigation links for Clients, Projects, and Invoices. The main form area has a light yellow background and contains the following fields:

- Invoice Number \*: A text input field containing "INV-2025-031".
- Project \* (Select project first): A dropdown menu showing "Select a project".
- Client \* (From selected project): A dropdown menu showing "Select a project first".
- Amount \*: An empty text input field.
- Date \*: A date input field with a calendar icon and placeholder "mm/dd/yyyy".
- Due Date \*: A date input field with a calendar icon and placeholder "mm/dd/yyyy".
- Status \*: A dropdown menu showing "Pending".
- Description: A large text area for notes.

At the bottom are two buttons: a blue "Create Invoice" button and a grey "Cancel" button.

## Backend work: (JSON Server)



# TESTING

## Functional Testing

- Verified CRUD operations work correctly
- Dashboard auto-updates after changes

## Unit Testing

- API endpoints tested manually using browser tools
- All pages load without errors

## UI Testing

- Checked responsiveness on laptop and mobile
- Verified color theme and spacing

## Performance Testing

- Vite build loads fast
- API responses from JSON Server are quick

## Advantages of the System

- Simple and fast CRM solution for freelancers
- Fully responsive interface
- Real-time updates via React state
- Clean and modern UI
- Lightweight backend using JSON Server

## Limitations of the System

- No user authentication
- No cloud database
- Invoices cannot be exported as PDF
- Multi-user login not supported

## FUTURE ENHANCEMENTS

- Add user login/signup
- Cloud backend using Firebase or MongoDB
- Invoice PDF generation
- Payment gateway integration
- Dark mode
- Analytics graphs using ChartJS
- Email reminders for overdue invoices

## **CONCLUSION**

The ClientSync project has been successfully developed as a comprehensive CRM system tailored for freelancers and small businesses. It integrates modern frontend technologies like React and Tailwind CSS with a lightweight backend powered by JSON Server, offering a smooth and efficient workflow. The system simplifies the management of clients, projects, and invoices by providing a centralized digital platform that is easy to use and visually appealing.

Throughout this project, I gained practical knowledge of component-based architecture, React hooks, routing, state management, and API integration. Implementing CRUD operations, responsive design, and real-time dashboard updates strengthened my understanding of full-stack web development.

ClientSync proves that even without a heavy backend, powerful applications can be built using the right combination of tools and architecture. With further enhancements like authentication, cloud storage, or PDF generation, this CRM system has the potential to evolve into a fully scalable professional product.

## **PROJECT DEMO LINK**

[https://drive.google.com/drive/folders/1Vhl3\\_igd\\_ZbRtvqZcDSEph9l0OFCtxis?usp=sharing](https://drive.google.com/drive/folders/1Vhl3_igd_ZbRtvqZcDSEph9l0OFCtxis?usp=sharing)

## **THANK YOU**