Raynex solar lead management

A PROJECT REPORT

Submitted by

RAJNIKANT HIRAPARA

92200133013

ABHAY PADARIYA

92200133017

BACHELOR OF TECHNOLOGY

in

Information And Communication Technology

Marwadi University, Rajkot

May, 2024

Information and Communication Technology

2023-24

CERTIFICATE

This is to certify that the project entitled Raynex solar lead management has been carried out by RAJNIKANT HIRAPARA (92200133013) and ABHAY PADARIYA (92200133017) under my guidance in partial fulfilment of the degree of Bachelor Engineering in Information and Communication Technology (6th Semester) of Marwadi University, Rajkot during the academic year 2024-25

Date : 17-05-2025

Internal Guide   **Head of the Department**

Mr. Shamsagazarzoo Alam   Prof . C. D. Parmar

Subject Co-ordinator   Head of Department ICT Engineering

C.T.O. Ally Soft Solutions

Abstract

The project titled Solar Lead Management is a web-based application designed to simplify and automate the process of managing customer leads in the solar energy sector. As the demand for solar power increases, businesses require an efficient system to track and categorize potential clients from initial enquiry to project completion.

This platform enables administrators to add new leads by filling out a detailed project creation form, capturing essential information such as customer details, project size, lead source, status lane, and type. Once created, these leads are dynamically displayed on the dashboard under categorized sections—Enquiry, Site Visit, Follow-Up, and Complete—based on their current status in the workflow.

The frontend is developed using React.js with Vite, providing a fast, modular, and highly responsive user experience. The backend is powered by Node.js and Express.js, handling all server-side logic and API integration. The data is stored securely in MongoDB Atlas, a cloud-based NoSQL database that supports scalability and real-time access.

Key features include dynamic form input, categorized lead visualization, seamless data flow between frontend and backend, and a clean, intuitive user interface. This system not only improves lead organization and communication but also empowers solar companies to optimize their sales pipeline, reduce manual effort, and enhance decision-making through better data visibility.

Index

CHAPTER 1

DESCRIPTION

1.1 Project Summary

The **Solar Lead Management System** is a web-based application designed to manage and track solar project leads efficiently. It helps users organize leads through different stages such as Pre-Qualified, Enquiry, Quotation, Design, and Installation. Built using React.js (Vite) for the frontend, Node.js with Express for the backend, and MongoDB Atlas for cloud data storage, the system ensures speed, scalability, and security. Key features include user authentication, project and task management, and a clean dashboard interface with sections for Projects, Tasks, and Settings. This platform streamlines the lead handling process, boosts team productivity, and keeps all project updates in one place.

1.2 Purpose

The primary purpose of the Solar Lead Management System is to digitize and automate the process of handling solar project leads, addressing common issues such as poor organization, missed follow-ups, and lack of visibility. This system provides a structured workflow that guides leads from initial enquiry to final project completion. With an intuitive interface, users can easily add, manage, and track project details in real-time. Cloud-based data storage ensures that all information is secure and accessible from anywhere. By centralizing lead data and minimizing manual errors, the system enhances decision-making, improves sales conversions, boosts customer satisfaction, and increases overall operational efficiency for solar businesses.

1.3 Features

The Solar Lead Management System offers a range of features designed to enhance usability, efficiency, and functionality. At the core of the system is a categorized dashboard that organizes leads under clearly defined tabs such as Enquiry, Site Visit, Follow-Up, and Complete, making it easy for users to monitor project progress. The platform includes a "Create Project" feature, allowing users to add new leads by entering essential details like customer name, contact information, project type, and current status. Navigation is simple and intuitive—clicking the logo brings users back to the main dashboard, while the "Create Project" button opens a dedicated form for adding new leads. The application is built with a

responsive design, ensuring smooth functionality across desktops, tablets, and mobile devices. Form validation is integrated to make sure all necessary information is submitted correctly. With real-time database integration using MongoDB Atlas, all project data is securely stored and updated instantly. Communication between the frontend and backend is handled efficiently through RESTful APIs, while the user interface is designed with Bootstrap, offering a clean and professional experience for all users.After placing an order, customers are directed to the Order page, where they can track the status of their purchase. This page shows payment confirmation, expected delivery dates, and a list of ordered products, giving customers full visibility into their transactions. With these comprehensive features, the e-commerce platform ensures a convenient and reliable shopping experience for all users.

1.4 Technology

- ➤ **Frontend Technology**

- Html
- CSS
- JavaScript
- React.js

- ➤ **Backend Technology**
- Node.js
- Express.js
- Mongo DB Atlas

1.5 Tools

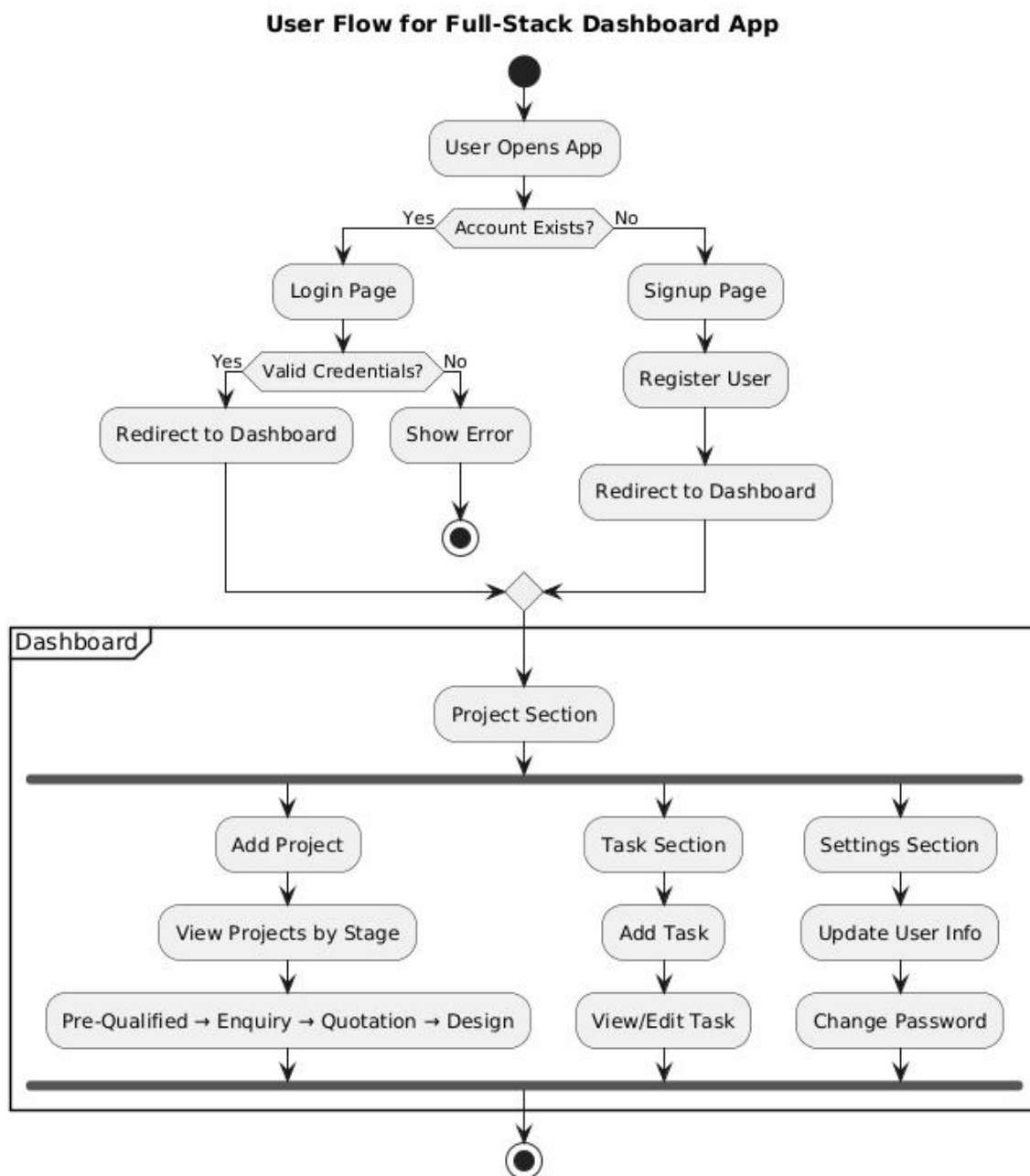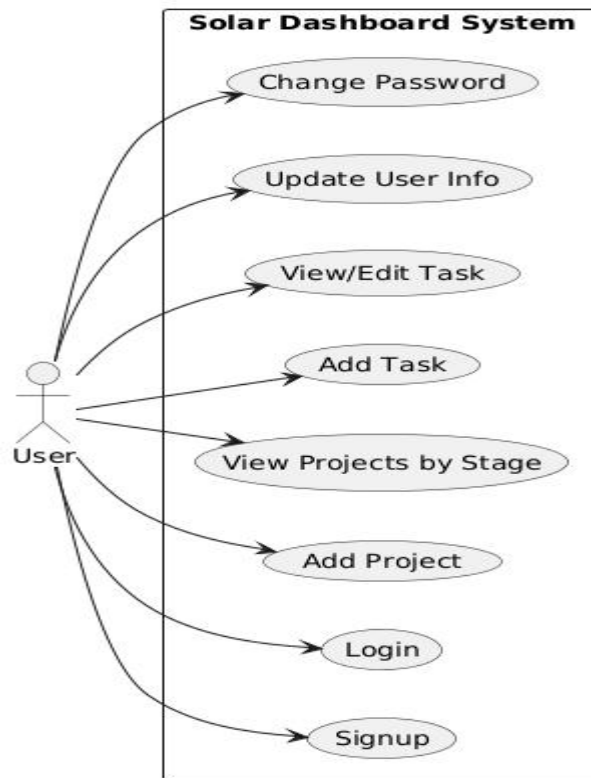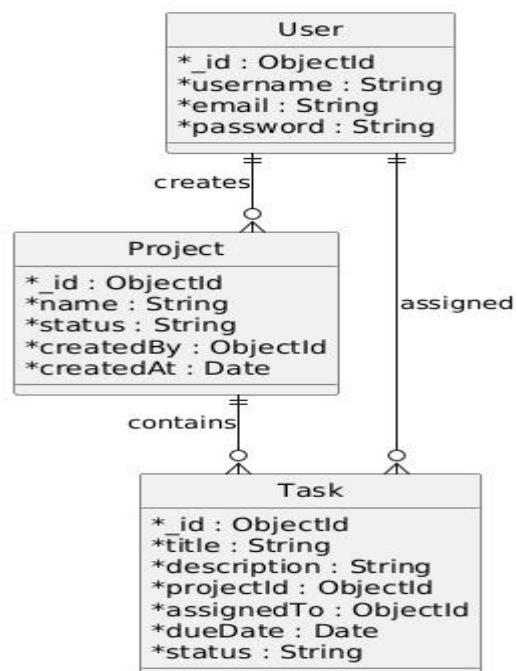- VS code
- Command prompt

CHAPTER 2

DIAGRAM

2.1 Flow Chart

User Flow for Full-Stack Dashboard App

Fig 2.1 Website Flowchart

2.2 Use case Diagram

2.3 ER-Diagram

CHAPTER 3

CODE SNIPPET & SCREENSHOTS

3.1 Login.js

```javascript
import React, { useState } from 'react';
import './login.css';
import { useNavigate } from 'react-router-dom';

const LoginPage = () => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const navigate = useNavigate();

  const handleLogin = async (e) => {
    e.preventDefault();
    try {
      const response = await fetch('http://localhost:5000/login', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({ email, password }),
      });
      const data = await response.json();
      if (data.success) {
        alert('Login successful!');
        // Optionally save token or user data
        // localStorage.setItem('token', data.token);
        navigate('/Dashboard/Projects');
      } else {
        alert('Invalid email or password');
      }
    } catch (error) {
      console.error('Login failed:', error);
    }
  };
  return (
    <div className="login-container">
```

3.2 App.js

```
import React from 'react';
import './App.css';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';

import Header from './components/Header';
import DashboardLayout from './components/DashboardLayout';
import Navbar from './components/Navbar';
import Hero from './components/Hero';
import About from './components/About';
import Services from './components/Services';
import WorkingProcess from './components/WorkingProcess';
import Contact from './components/Contact';
import Footer from './components/Footer';
import Login from './components/login';
import ForgotPasswordPage from './components/ForgotPasswordPage';
import SignUpPage from './components/SignUpPage';
import WhySolar from './components/why-solar';
import Dashboard from './components/Dashboard';
import Projects from './components/Projects';
import Tasks from './components/Tasks';
import Settings from './components/Settings';
import Payments from './components/Payments';
import Leads from './components/Leads';
import Books from './components/Books';
    <Route path="/" element={
        <>
            <Header />
            <Navbar />
            <Hero />
            <About />
            <Services />
            <WorkingProcess />
            <Contact />
            <Footer />
        </>
    } />
```

3.3 db.js

```js
const mongoose = require('mongoose');

// MongoDB connection string
const mongoURI = 'mongodb://localhost:27017/solarApp'; // Or use MongoDB Atlas
connection string

const connectDB = async () => {
  try {
    // Only attempt connection if not already connected
    if (mongoose.connection.readyState === 0) {
      await mongoose.connect(mongoURI, {
        useNewUrlParser: true,
        useUnifiedTopology: true,
      });
      console.log('MongoDB Connected...');
    } else {
      console.log('MongoDB already connected.');
    }
  } catch (err) {
    console.error('❌ MongoDB Connection Error:', err.message);
    process.exit(1);
  }
};

module.exports = connectDB;
```

3.4 UserRoutes.js

```js
const express = require('express');
const router = express.Router();

// POST /api/user/settings
router.put('/settings', (req, res) => {
  const settings = req.body;
  console.log('Received settings:', settings);

  // Here you could save settings to DB
  res.json({ success: true, message: 'Settings saved successfully' });
```

3.5 Server.js

```javascript
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');
const multer = require('multer');
const path = require('path');
const dotenv = require('dotenv');

const projectRoutes = require('./routes/projects');
const userRoutes = require('./routes/userRoutes');
const tasksRouter = require('./routes/Tasks');
dotenv.config();

const app = express();
app.use(cors());
app.use(express.json());
app.use(express.static('uploads'));

const mongoURI =
'mongodb+srv://rajnikanthirpara8200:meetrajni@cluster1.jlxmhzu.mongodb.net/solar-
db?retryWrites=true&w=majority&appName=Cluster1';
mongoose.connect(mongoURI, {
  useNewUrlParser: true,
  useUnifiedTopology: true,
}).then(() => console.log('✅ Connected to MongoDB Atlas'))
  .catch((err) => console.error('❌ MongoDB connection error:', err));

const userSchema = new mongoose.Schema({
  username: String,
  email: String,
  password: String,
  profilePicture: String,
});

const User = mongoose.model('User', userSchema);

const storage = multer.diskStorage({
  destination: (req, file, cb) => {
    cb(null, './uploads/profile-pictures')
```
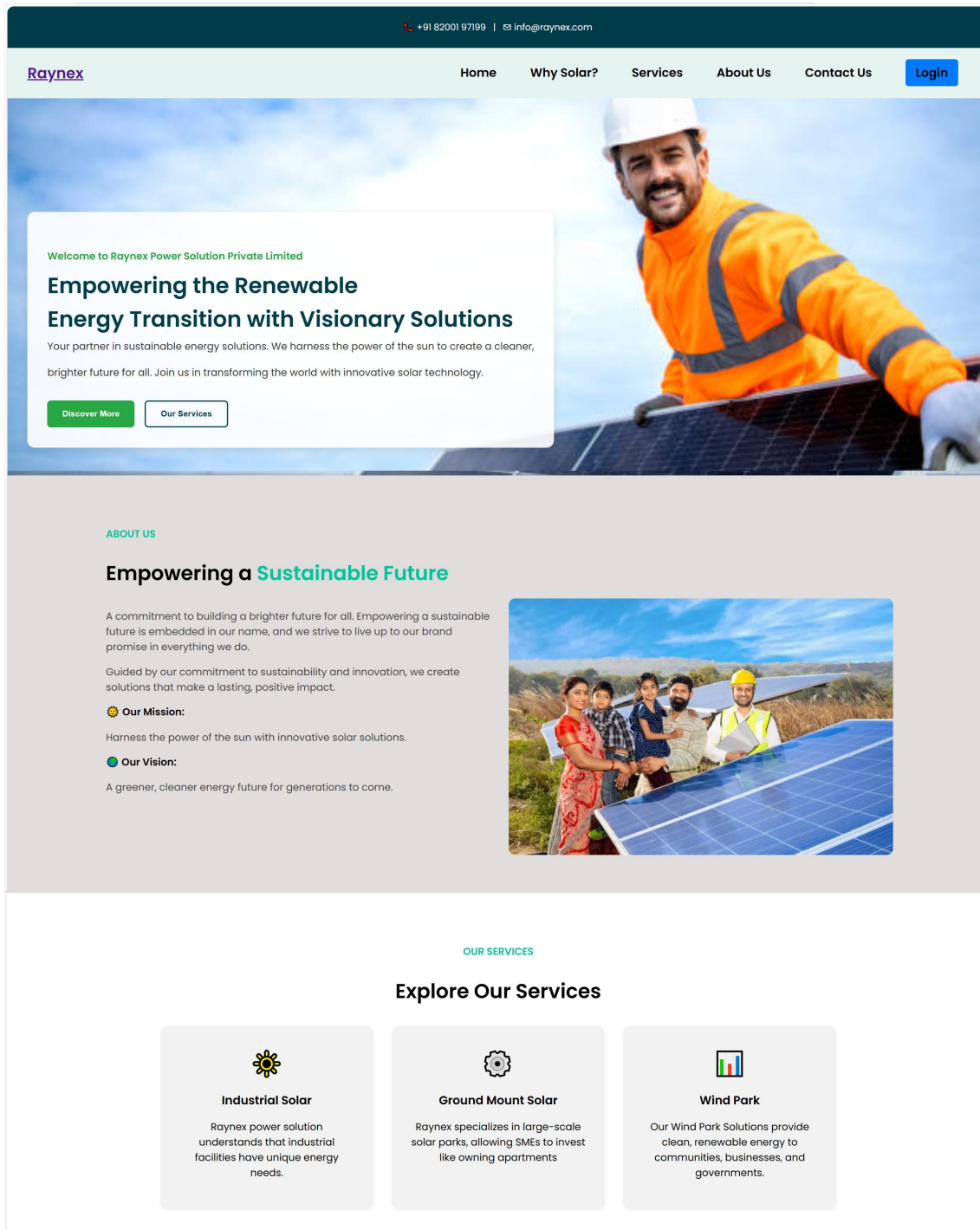
3.6 Output:

OUR PROCESS

Working Process for Services

Our streamlined process ensures seamless service from consultation to installation. We design, install, and maintain your solar solution with care.

Project Planning

We assess your energy needs, design custom solar solution, and manage installation for optimal efficiency.

Research & Analysis

We conduct in-depth research and analysis to design efficient solar solutions tailored to your needs.

Solar Installation

Professional solar installation with precision & care, ensuring optimal performance & long-term energy savings.

Ready To Use

Grid tariffs increase by 3-5% annually, driving up electricity costs. Switch to solar and secure long-term savings!

CONTACT US

Get in Touch

| Your Name |
| Your Email |
| Your Message |

Send Message

Raynex Power Solution Pvt. Ltd.

📍 806, Sarthana Business Hub, near Taxshila classes, surat-395006

📞 +91 82001 97199

✉ info@raynex.com

Login

| Email Address |
| Password |

Login

Forgot Password? | New User? Sign Up

Sign Up

| Username |
| Email Address |
| Password |

Sign Up

Already have an account? Login

Raynex Solar

Projects
Tasks
Settings

Logout

Search projects...   + Add Project

Pre Qualified

rajnikant   Sat May 24 2025
                       commercial
ray-01   raynex   5 KW

meet   Fri May 16 2025
                       industrial
ray-02   raynex   3 KW

Enquiry

vraj   Sun May 18 2025
                       industrial
ray-04   raynex   3 KW

rajni   Sat May 24 2025
                       residential
ray-06   raynex   12 KW

Quotation

abhay   Sat May 17 2025
                       residential
ray-03   raynex   5 KW

Design

harikrushna   Sun May 25 2025
                       residential
ray-05   raynex   10 KW

Raynex Solar

Projects
Tasks
Settings

Tasks

+ Add Task

Task	Due	Customer	Assignee	Status
☐ this and this rynex	30/5/2025, 4:25:00 am	harikrushna	solar water	Completed
☐ complete this task rynex	31/5/2025, 12:41:00 am	harikrushna	solar water	In Progress
☐ mmmmm rynex	24/5/2025, 4:42:00 am	harikrushna	solar water	Pending
☐ complete solar plant rynex	24/5/2025, 5:23:00 am	harikrushna	solar water	Pending

---

Raynex Solar

Projects
Tasks
Settings

Logout

Settings

Profile Settings

Username:

Email:

Profile Picture:

Choose File No file chosen

Account Settings

Theme:

Light

Language Preference:

English

3.7 Future Enhancements:

In the future, the Solar Lead Management System can be enhanced with several advanced features to improve its functionality, scalability, and user experience. Role-based access control can be implemented to define different access levels for Admins, Sales Executives, and Managers. The system can include automatic lead assignment based on region, availability, or performance, as well as integration of email and SMS notifications for real-time lead updates and appointment reminders. A detailed analytics dashboard with visual reports can help track lead conversion rates and team performance. Additionally, a mobile application can be developed for on-the-go updates during site visits, along with Google Maps integration for route optimization. Features like document uploads, lead source tracking, chatbot support, and secure payment gateway integration for advance bookings can also be added. These enhancements will make the platform more efficient, user-friendly, and adaptable to growing business needs.

<u>References</u>

[1] https://www.youtube.com/watch?v=D389COfRSCM&t=1098s&pp=ygUVcGhwIGVjb21tZXJjZSB3ZWJzaXRl

[2] https://youtu.be/QNWIlrW7q_o?si=se0buLVavrket4M8

[3] https://www.youtube.com/watch?v=VhZh1xDwqV0

[4] https://github.com/mohsinenur/E-Commerce-Website-Using-PHP