Way To Success

MINOR PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR

THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

Information Technology

Submitted By:

Amisha Bagri(1805486)

Arshdeep Kaur(1805493)

Madvi(1805523)

Submitted To:

Prof. Ranjodh Kaur

Assistant Professor

Minor Project Coordinator

Department of Information Technology

Guru Nanak Dev Engineering College,

Ludhiana-141006

Abstract

Way To Success website is an advertising platform which focuses mainly on Small Scale Industries. This will help the industries to promote their products free of cost which help them to explore new opportunities and grow their business. The frontend is Developed using React JS and Sass (an extenstion of CSS) which helped the site to be more interactive for the user. The backend is based on Firebase which stores the user and the company data securely. We also provide added security to our users by authenicating them via mails and sms. We have also added dark theme mode which is surely in trend these days. The users can interact with various industries and can even use filters to narrow down their choices. The review and rating system alsonhelp the users to share their views and can help others to choose what's best for them.

In conlcusion, our site wants the business to grow and to help them broaden their horizons.

ACKNOWLEDGEMENT

WE are highly grateful to the Dr. Sehijpal Singh, Principal, Guru Nanak Dev Engineering College (GNDEC), Ludhiana, for providing this opportunity to carry out the minor project work at Way To Success.

The constant guidance and encouragement received from Dr. K.S.Mann, H.O.D., IT Department, GNDEC Ludhiana has been of great help in carrying out the project work and is acknowledged with reverential thanks.

WE would like to express a deep sense of gratitude and thanks profusely to Dr. Kiran Jyoti, without her wise counsel and able guidance, it would have been impossible to complete the project in this manner.

WE express gratitude to other faculty members of Information Technology Department of GNDEC for their intellectual support throughout the course of this work.

Finally, WE are indebted to all whosoever have contributed in this report work.

Amisha Bagri

Arshdeep Kaur

Madvi

List of Figures

.

List of Tables

Contents

1 Introduction

1.1 Introduction to Project Page

Way to Success website is a platform for the industries to advertise their products. This platform's main focus is SMALL SCALE INDUSTRIES. Various industrialist can join our initiative for free and the promotion of the product is free of cost.

This will be a Web based Application designed in REACT powered by Google Firebase. React is an open-source front-end Javascript Library for building user interfaces or UI components. Firebase is a Backend-as-a-Service(Baas).It provides the developers with a variety of tools and services to help them grow their user base.

The user of our platform can securely SIGNUP for our site as it is completely secure. The industries can look for the products they want and choose from the best. With the rating and the review system, user can actually decide what's best for them. The data of the user will be safely stored at the Firebase Backend. We have also given the user the filter facility where they can choose the specific industry they want to interact with.

Overall this is a platform that allows THE SMALL SCALE INDUSTRY to grow altogether and expand their reach. This will help them to interact with more people which will inreturn expand their horizon.

1.2 Project Category (Internet based, Application or System Development, Research based, Industry Automation, Network or System Administration)

This is a Internet Based Multiple Pages Web Application with a secured Backend.

1.3 Objectives

1. **PLATFORM TO PROMOTE THE PRODUCTS:** By registering the company on our site, the business people will be able to add products: their name, description and image which will help them to create a catalogue for the company and thus will provide them the platform to promote their products. This website is entirely free of cost .The companies only need to verify their mail id and phone number after which they are good to go. Since our site is perfectly authenticated and secured, the user will need not to worry about the privacy . They will receive OTPs through protected system and hence their data is perfectly safe .

2. **RATING AND REVIEW**: We also provide the facility to review and rate the services provided by the companies. In this way, a person can easily compare certain industries and can choose what's best for them. The review can only be given by the person who have verified their mail ids and phone number. In this way we can provide our consumers a genuine ratings.

1.4 Problem Formulation

We already know about platform that already provide such service. They provide a vast network for the promotion of their products. But the one major problem with these sites is that they charge money for the services they provide. Very limited Small Scale Industries can avail such opportunities as some prefer that its not feasible for them to spend such amount of money on promotion. So in conclusion, the Large Scale Industries are only benefited from the already available system and as such no digital platform is available for the Small Businesses.

1.5 Identification/Reorganization of Need

Since Ludhiana is a hub of industries, most of the families here own one or another kind of business. So we are pretty familiar with the problems faced by the small scale industries in this sector. We have seen our families paying like Rs 40,000-Rs 60,000 to such platforms in order their promote products. But after even paying such amount of money, the outcome is not that fruitful. Business owners they get queries from places which are very far way and those companies who contact them are not ready to pay for all such long distance arrangements. So in this way all the time is wasted. Since we have seen this problem closely, we are able to provide a better solution for this.

1.6 Existing System

For existing system we have example of TRADE INDIA and INDIA MART. So lets take an example, if you go to these sites and you try to find a product, you will get suggestions for it from a place that's like thousand miles away and its not affordable for you to import that product. So in this case rather than giving you the facility which might be near around you they provide suggestions which are not useful for you. In this way, you the consumer and the manufacturer of the service both are losing. The consumer is not happy with the suggestion and the manufacturer or the service provider who spent thousands for the promotion of their products don't end up getting the order. So we can say the existing platform is not that reliable.

1.7 Proposed System

Since we had already gone through the drawbacks of the existing system, we can conclude they are more feasible for LARGE BUSINESSES.

But the main focus of our site is SMALL SCALE INDUSTRIES. We are providing them a free, reliable and a secured platform to promote their products. On our platform the manufacturers, service providers and consumers they can come altogether to help each other grow. On our site the company can register themselves and display their products. We also give the power to the consumers to review and rate the services provided by the companies. This thing will help not only the future buyers but also give the opportunity to the companies to improve themselves.

1.8 Unique Features of the System

- **Secured Site:** Our site is fully secured and hence our consumers can easily trust us with their private information.
- **Phone and Mail Verifications**: We authenticate our users by sending them verification OTPs on the registered email and phone number.
- **Responsive**: our site fully responsive and it can be viewed on any browser
- **Password using hashing**: The password set by the users are not visible to us as we use hashing, thus providing them fully privacy.
- **Dark Mode:** The toggle button on the right side will help the user to toggle between light and dark mode.

2 Requirement Analysis and System Specification

2.1 Feasibility Study (Technical, Economical, Operational)

A feasibility analysis evaluates the project's potential for success; therefore, perceived objectivity is an essential factor in the credibility of the study for potential investors and lending institutions. There are three types of feasibility study—separate areas that a feasibility study examines, described below.

1. **Technical Feasibility**: - This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves evaluation of the hardware, software, and other technical requirements of the proposed system. This project is technically feasible.

2. **Economic Feasibility**: This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide. This project is economically feasible as the financial requirements of this project is minimal as it only requires a server.

3. **Operational Feasibility:** This assessment involves undertaking a study to analyze and determine whether and how well the organization's needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development. This project is operationally feasible as it provides several benefits in one product.

2.2 Software Requirement Specification Document which must include the following: (Data Requirement, Functional Requirement, Performance Requirement, Dependability Requirement, Maintainability requirement, Security Requirement, Look and feel requirement)

Programming:

- React
- React Native

- JavaScript

Tools:

- VS Code (Ide)
- Node Js
- Chrome Dev Tools
- Firebase

Supporting Dependences:

- Redux
- Node-Sass
- Material UI
- Firebase-Tools
- React Router Dom

Package Manager:

- Npm: Node Package Manager
- Yarn: Yet Another Resource Negotiator

Hardware Requirement

- Operation System: Windows 7 or higher, MAC, Linux.
- Processor: 2.80GHz.
- Ram: 8-GB.
- Display: 1920 X 1080 resolution.
- Ethernet connection or a wireless adapter.

2.3 Validation

Validation is the documented process of demonstrating that a system or process meets a defined set of requirements. There are a common set of validation documents used to provide this evidence. A validation project usually follows this process:

- **Validation Planning** – The decision is made to validate the system. A project lead is identified, and validation resources are gathered.
- **Requirement Gathering** – System Requirements are identified. Requirements are documented in the appropriate specifications. Specification documents are reviewed and approved.
- **System Testing** – Testing Protocols are written, reviewed, and approved. The protocol is executed to document that the system meets all requirements.
- **System Release** – The Summary Report is written and system is released to the end-users for use.
- **Change Control** – If changes need to be made after validation is complete, Change Control ensures that the system changes does not affect the system in unexpected ways.

2.4 Expected hurdles

We need to advertise our platform so that companies can get to know about us and also we require some initial database to start with.

2.5 SDLC Model to be used

Incremental Model is a process of software development where requirements divided into multiple standalone modules of the software development cycle. In this model, each module goes through the requirements, design, implementation and testing phases. Every subsequent release of the module adds function to the previous release. The process continues until the complete system achieved.

1. **Requirement analysis**: In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.
2. **Design & Development:** In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.
3. **Testing:** In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, the various methods are used to test the behavior of each task.
4. **Implementation:** Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in

the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product

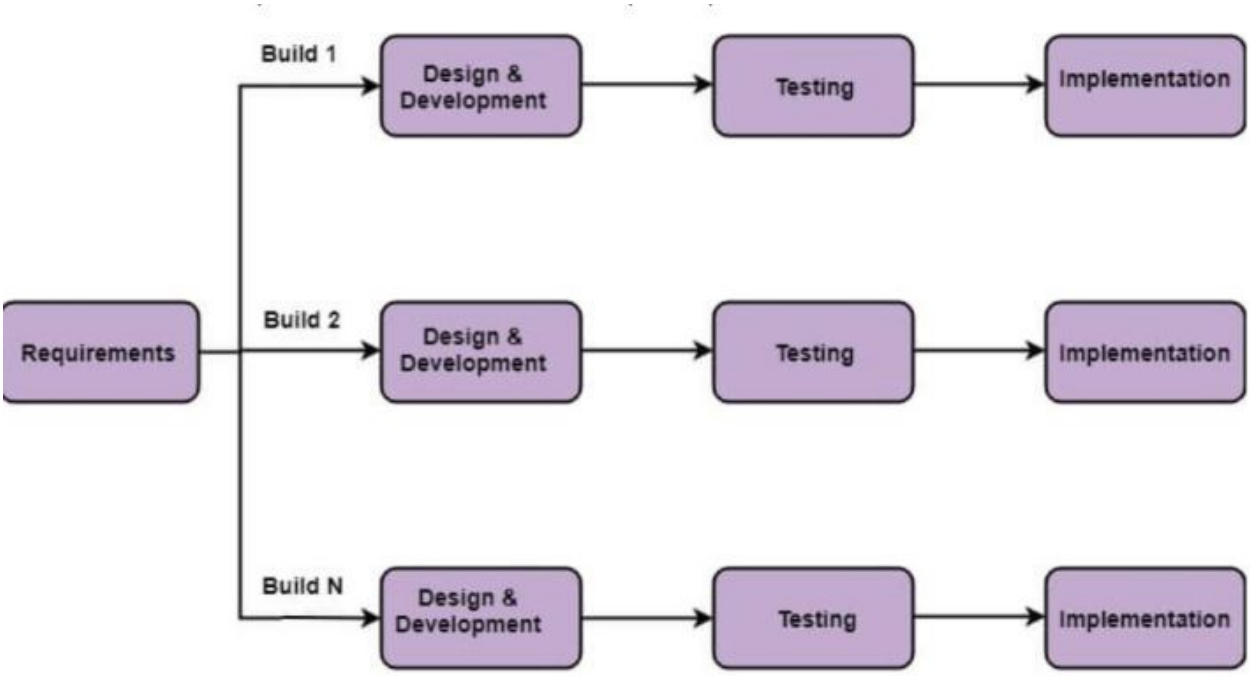


Figure 1: SDLC MODEL

3 System Design

3.1 Design Approach

WORK FLOW

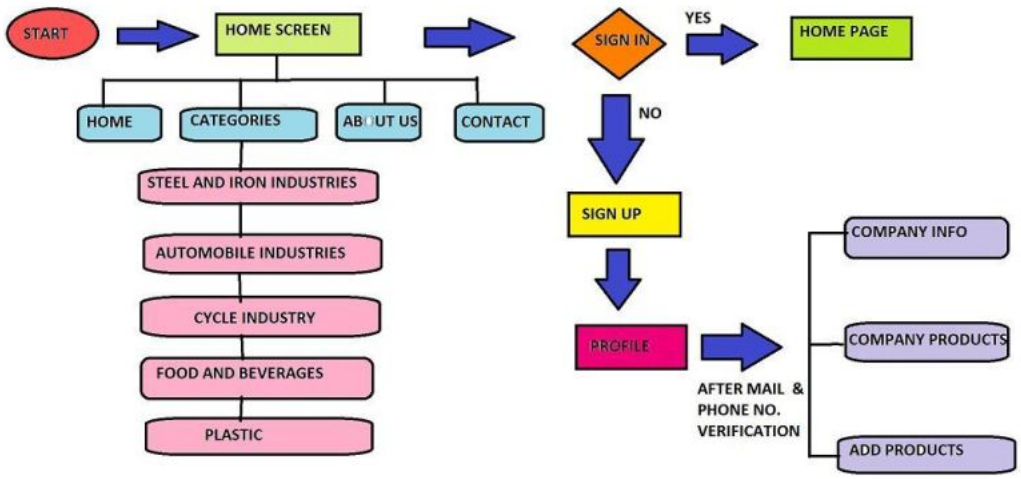


Figure 2: WORK FLOW

CORE MODULE

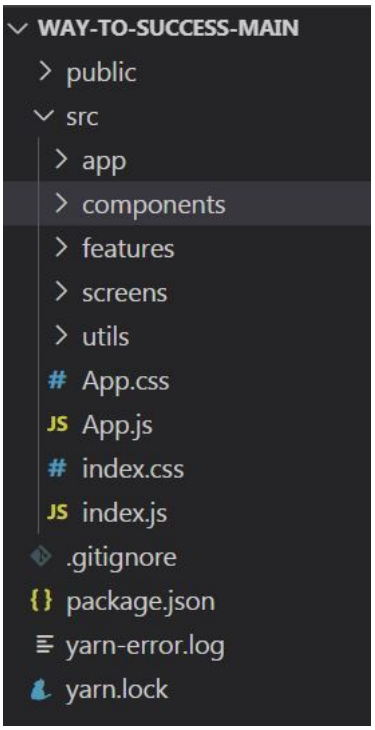


Figure 3: CORE MODULE

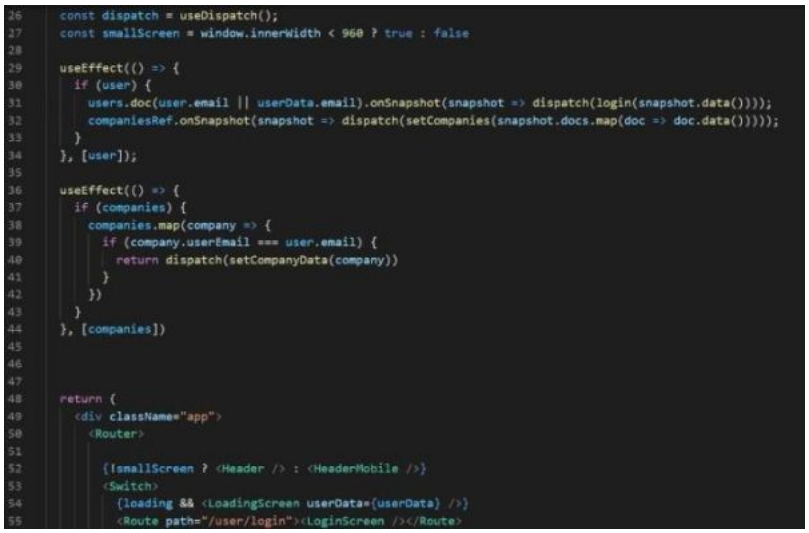
Since our project contains lots of coding portion, we are going to show the main portion which binds everything togther i.e APP.JS

```
import React, { useEffect } from 'react';
import './App.css';
import { useAuthState } from 'react-firebase-hooks/auth';
import { BrowserRouter as Router, Switch, Route } from "react-router-dom";
import { auth, companies as companiesRef, users } from './utils/firebase';
import HomeScreen from './screens/HomeScreen';
import LoadingScreen from './screens/LoadingScreen';
import LoginScreen from './screens/LoginScreen'
import RegisterScreen from './screens/RegisterScreen';
import AboutUs from './screens/AboutUs';
import ContactUs from './screens/ContactUs';
import Dashboard from './screens/Dashboard';
import { useDispatch, useSelector } from 'react-redux';
import { login, selectUserData, setCompanyData, selectCompanies, setCompanies } from './features/appSlice';
import CategoryScreen from './screens/CategoryScreen';
import Header from './components/Header';
import HeaderMobile from './components/HeaderMobile';
import AdminBlock from './screens/AdminBlock';
import Company from './screens/Company';
import AppUser from './screens/AppUser';

function App() {
  const [user, loading] = useAuthState(auth);
  const userData = useSelector(selectUserData);
  const companies = useSelector(selectCompanies);
  const dispatch = useDispatch();
  const smallScreen = window.innerWidth < 960 ? true : false

  useEffect(() => {
    if (user) {
```

Figure 4: APP SCREEN JS



```
const dispatch = useDispatch();
const smallScreen = window.innerWidth < 960 ? true : false

useEffect(() => {
  if (user) {
    users.doc(user.email || userData.email).onSnapshot(snapshot => dispatch(login(snapshot.data())));
    companiesRef.onSnapshot(snapshot => dispatch(setCompanies(snapshot.docs.map(doc => doc.data()))));
  }
}, [user]);

useEffect(() => {
  if (companies) {
    companies.map(company => {
      if (company.userEmail === user.email) {
        return dispatch(setCompanyData(company))
      }
    })
  }
}, [companies])



return (
  <div className="app">
    <Router>

      {!smallScreen ? <Header /> : <HeaderMobile />}
      <Switch>
        {loading && <LoadingScreen userData={userData} />}
        <Route path="/user/login"><LoginScreen /></Route>
```

Figure 5: APP SCREEN 2

```
51
52          {!smallScreen ? <Header /> : <HeaderMobile />}
53          <Switch>
54            {loading && <LoadingScreen userData={userData} />}
55            <Route path="/user/login"><LoginScreen /></Route>
56            <Route path="/user/register"><RegisterScreen /></Route>
57            <Route path="/user/dashboard" ><Dashboard /></Route>
58            <Route path="/categories/:id" ><CategoryScreen /></Route>
59            <Route path="/contact" ><ContactUs /></Route>
60            <Route path="/about" ><AboutUs /></Route>
61            <Route path="/admin" ><AdminBlock /></Route>
62            <Route path="/company/:id" ><Company /></Route>
63            <Route path="/appuser/:id" ><AppUser /></Route>
64            <Route path="/" exact><HomeScreen /></Route>
65          </Switch>
66        </Router>
67      </div>
68    );
69  }
70
71  export default App;
```

Figure 6: APP SCREEN 3

3.2 Detail Design

- **Front-End**

The Front End is developed using React and JavaScript libraries which is integrated with Firebase Backend. We even used SASS i.e. an extension of CSS which helps to make our site even more presentable and add colours to it. The dark and Light theme mode also help the user to use our platform comfortably which enhances user experience.

- **Back-End**

We use firebase as a backend which is a very powerful tool. This tool provides authentication, authorization, and managing a realtime database.It works seamlessly and helps to reduce development hours. This is very safe for users as it uses hashing and their password details are not share with the owners.

- **Data-Base**

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

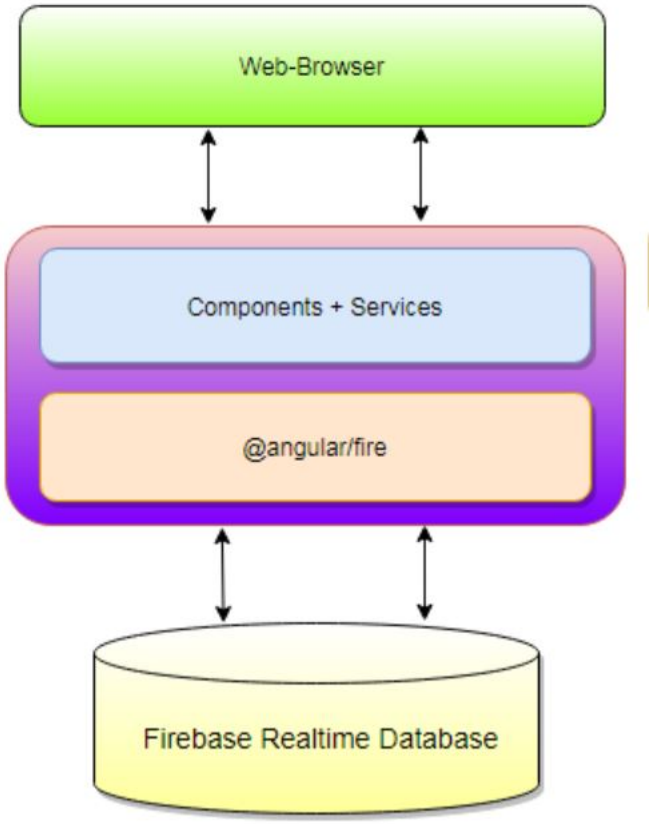


Figure 7: FIREBASE

3.3 Data Flow

Figure 8: DATA FLOW 1

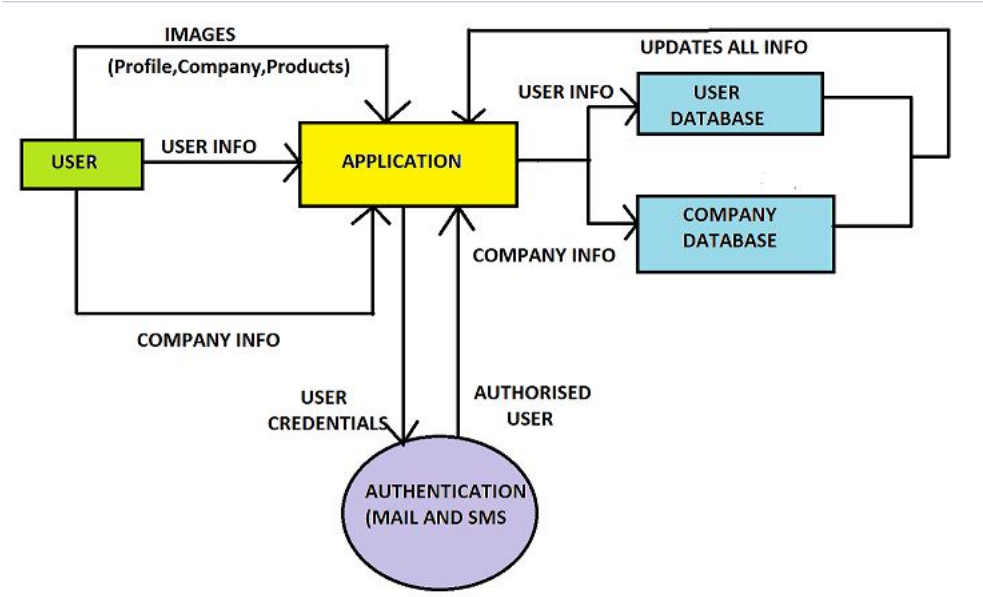


Figure 9: DATA FLOW 2

3.4 Database Design

NoSQL databases are designed to break away from the rows and columns of the relational database model. But it's a common mistake to think that NoSQL databases don't have any sort of data model. A useful description of how the data will be organized is the beginning of a schema. Relational databases have had generations of users and developers to work out standard design methods. Various formal tools exist for describing the relationships between the main objects in a business domain, and these formal descriptions can then be used to dictate how the data will be stored. The same types of standard data modeling tools are not available for NoSQL data modeling. One recommendation is to begin with a business domain model expressed in a form that can be incorporated in an application, such as a JSON document. Another important design driver is the types of data access that need to be supported. Some use cases require access via a query language and others require access by one or more applications. Because no business or application domain is static, change over time must also be taken into account. When it comes to flexibility, NoSQL database schemas are much less costly to revise.

3.4.1 ER Diagrams

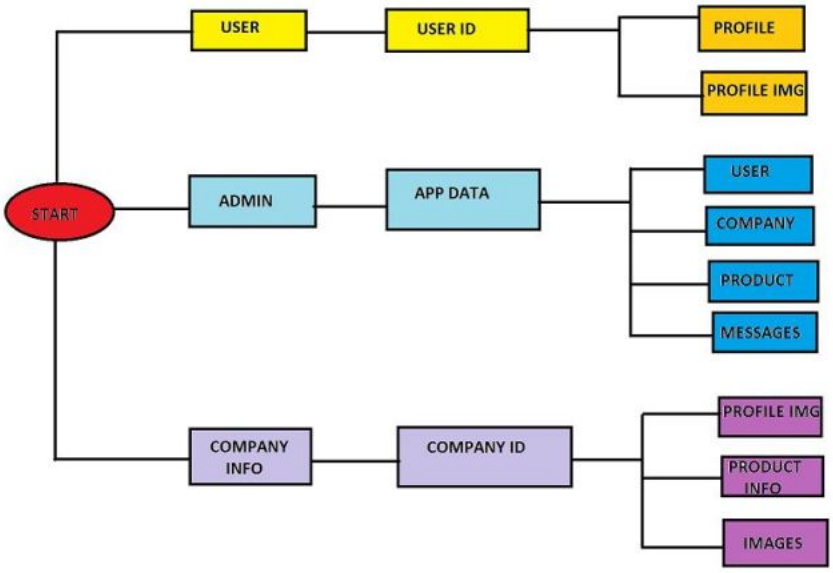


Figure 10: ER DIAGRAM

4 Implementation, Testing, and Maintenance

4.1 Introduction to Languages, IDE's, Tools and Technologies used for Implementation

- **JAVASCRIPT** :JavaScript is one of the most used programming languages in the web development field. JavaScript is one of the easiest languages that is combined with the HTML to create dynamic web documents. JavaScript is available to all and it is the language which is cross platform and runs on every system. JavaScript is mostly used to develop frontend and backend application in web domain. JavaScript doesn't need any special platform as it is easily available and runnable on every code editor.

- **React JS**: React is a JavaScript library for building user interfaces. It allows us to create reactive interfaces very easily because it can listen for state change and when state change occurs, it will update UI layout. Its component based and it uses XML-like syntax called JSX. React is use to create SPA (Single Page Application) which make it fast and responsive. React is not a framework. It is just a library developed by Facebook to solve some problems that we were facing earlier.

- **REDUX:** Redux is a predictable state container designed to help you write JavaScript apps that behave consistently across client, server, and native environments and are easy to test. While it's mostly used as a state management tool with React, you can use it with any other JavaScript framework or library. It's lightweight at 2KB (including dependencies), so you don't have to worry about it making your application's asset size bigger. Lately one of the biggest debates in the frontend world has been about Redux. Not long after its release, Redux became one of the hottest topics of discussion. Many favoured it while others pointed out issues.

- **JSX**: JSX is an XML-like syntax extension to ECMAScript without any defined semantics. JSX allows us to write HTML elements in JavaScript and place them in the DOM without any createElement() and/or appendChild() methods. It's NOT intended to be implemented by engines or browsers. Basically, by using JSX you can write HTML code in JavaScript then Babel transforms these expressions into actual JavaScript code. It allows us to put HTML into JavaScript. JSX is like a shorthand for calling React.createElement function.

Other Supporting tools :

- **NODE JS** :Since JavaScript is limited to the frontend, we can add JavaScript only to the browser but now we can run JavaScript on the server site. Node.js is a JavaScript runtime environment i.e., the environment where we can run the JavaScript and this is built on Chrome V8 JavaScript engine (It's

an engine built by chrome using the C++ and this engine is used by the chrome as well). This engine converts the JavaScript code to the machine code which can be understand by computer's microprocessor. Node.js can also be used to build the desktop applications. Using Node.js we can read the content of the file, create files, delete files and do all the file management that we want.

- **NPM** :npm is the package manager for the Node JavaScript platform. It puts modules in place so that node can find them, and manages dependency conflicts intelligently. It is extremely configurable to support a wide variety of use cases. Most commonly, it is used to publish, discover, install, and develop node programs. npm is the package manager for the Node JavaScript platform. It puts modules in place so that node can find them, and manages dependency conflicts intelligently. It is extremely configurable to support a wide variety of use cases. Most commonly, it is used to publish, discover, install, and develop node programs.

- **YARN** :YARN is an Apache Hadoop technology and stands for Yet Another Resource Negotiator. YARN is a large-scale, distributed operating system for big data applications. The technology is designed for cluster management and is one of the key features in the second generation of Hadoop, the Apache Software Foundation's open-source distributed processing framework. YARN is a software rewrite that is capable of decoupling MapReduce's resource management and scheduling capabilities from the data processing component.

- **FIREBASE**: Firebase is a Backend-as-a-Service — BaaS — that started as a YC11 start-up and grew up into a next generation app-development platform on Google Cloud Platform. Firebase frees developers to focus crafting fantastic user experiences. You don't need to manage servers. Real-time Database: When you connect your app to Firebase, you're not connecting through normal HTTP. You're connecting through a WebSocket. WebSocket's are much, much faster than HTTP. You don't have to make individual WebSocket calls, because one socket connection is plenty. All of your data syncs automagically through that single WebSocket as fast as your client's network can carry it.

File Storage: Firebase Storage provides a simple way to save binary files — most often images, but it could be anything — to Google Cloud Storage directly from the client. Firebase Storage has its own system of security rules to protect your Google Cloud bucket from the masses, while granting detailed write privileges to your authenticated clients.

Authentication: Firebase authentication has a built-in email/password authentication system. It also supports OAuth2 for Google, Facebook, Twitter and GitHub. We'll focus on email/password authentication for the most part.

Hosting: Firebase includes an easy-to-use hosting service for all of your static files. It serves them from a

global CDN (content delivery network) with HTTP/2. Dependences Lot of dependences are used in this project

Few of them are:

• **Material UI**: Material UI is a component library for React teeming with powerful components that you should be using in your projects. If you're just looking to create a good-looking app, Material UI can provide you with solid pre-styled components that will get the job done.

• **React Router Dom**: A tool that allows you to handle routes in a web app, using dynamic routing. Dynamic routing takes place as the app is rendering on your machine, unlike the old routing architecture where the routing is handled in a configuration outside of a running app. React router implements a component-based approach to routing.

• **UUID**: UUIDs are generally used for identifying information that needs to be unique within a system or network thereof. Their uniqueness and low probability in being repeated makes them useful for being associative keys in databases and identifiers for physical hardware within an organization.

• **Firebase:** Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure. Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents.

4.2 Coding standards of Language used

1. Naming conventions for variables, constants and functions:

• Meaningful and understandable variables name help anyone to understand the reason of using it.

• Variables should be named using camel case lettering starting with small letter (e.g., localData). Constant names should be formed using capital letters only (e.g. CONSDATA).

• It is better to avoid the use of digits in variable names.

• The names of the function should be written in camel case starting with small letters.

• The name of the function must describe the reason of using the function clearly and briefly. 2. Proper indentation is very important to increase the readability of the code. For making the code readable, programmers should use White spaces properly.

Some of the spacing conventions are given below:

• There must be a space after giving a comma between two function arguments.

• Each nested block should be properly indented and spaced.

• Proper Indentation should be there at the beginning and at the end of each block in the program

• All braces should start from a new line and the code following the end of braces also start from a new line.

3. Lengthy functions are very difficult to understand. That's why functions should be small enough to carry out small work and lengthy functions should be broken into small ones for completing small tasks

4.3 Project Scheduling using various tools such as GANTT charts

GANTT CHART

Table 1: GANTT CHART

ACTIVITY	STARTING DATE	ENDING DATE	DURATION	STATUS
IDEA GENERATION	28/02/2021	01/03/2021	1 Day	Completed
IDEA VALIDATION	02/03/2021	02/02/2021	Same Day	Completed
UI FOR LOGIN and REGISTER	04/03/2021	07/03/2021	3 Days	Completed
UI FOR HOME PAGE	08/03/2021	10/03/2021	2 Days	Completed
SCREEN CODING	11/03/2021	15/03/2021	4 Days	Completed
REPO DESIGNING	16/03/2021	18/03/2021	2 Days	Completed
UI for NAVIGATION PANE	18/03/2021	20/02/2021	2 Days	Completed
DATABASE CREATION	21/03/2021	23/03/2021	2 Days	Completed
INTEGRATING FIRE STORE	24/03/2021	27/03/2021	3 Days	Completed
UPLOADING SECTION	02/04/2021	04/04/2021	2 Days	Completed
ADMIN BLOCK	05/04/2021	07/04/2021	2 Days	Completed
REVIEW and RATING SYSTEM	08/04/2021	10/04/2021	2 Days	Completed
DARK THEME IMPLEMENTATION	11/4/2021	12/4/2021	1 Days	Completed

4.4 Testing Techniques and Test Plans

1. **Functionality Testing** The main goal of functional testing is to make sure that all the functions within a web app are working smoothly without any technical glitches. Our web app is working very smoothly without any type of glitch.

2. **Testing your responsive web designs** Every modern web designer should know the principles of responsive web design – how to get your sites to render perfectly on whatever device or screen size it appears on. Our web app is highly responsive and set itself according to the screen size or device.

3. **Usability Testing** When it comes to make the application user-friendly and effective, its user interface should comply with the standards.

4. **Web UI Testing** One of the most important interfaces within a web application are web server and application server interface and database server interface.

5. **Compatibility Testing** Compatibility of our web application is one of the most crucial things we should consider while testing the application. Compatibility testing will check our website or web application for browser compatibility, operating system compatibility, mobile browsing and printing options.

6. **Performance Testing** Performance testing will help us determine the performance of our web application under various scenarios.

7. **Security Testing** This testing method is one of the most important ones for our web application as if data leaks or modifications are tolerable or not. Our web app is completely secure website.

5 Results and Discussions

5.1 User Interface Representation (of Respective Project)

5.1.1 Brief Description of Various Modules of the system

Home Screen: This module contains a navigation to different screens like category screen, contact us screen etc. and it itself presents the brief view of different industries registered themselves with some information about team members and a brief about section of the site.

Login Screen: This is just to get logged in our website to visit some of the restricted portions. Register Screen This consists of signing up a new user to our web app. Category Screen This is a view of all the categories with their respective industries registered.

About Us Screen: This contains the little description of what our website provide to our users.

Contact Us Screen:This is the way how a user can give or provide their messages to the web app admins.

Admin Screen: All the activities happening on app record in admin screen. Users are also managed from this Screen. Only Admin and Moderators have access to this site of application. All the info about the companies,users,categories and messages record is there itself.

Dashboard Screen: This Dashboard consists of all about our info and the info of our company including the verifications through or gmail and SMS verification proceeding with the additions of addition of products with their images.

Company Screen: This is a representable form of any company with their required info about products and a slide show with the ratings and review by the other users.

5.2 Snapshots of system with brief detail of each

WEB APPLICATION

HOME SCREEN:

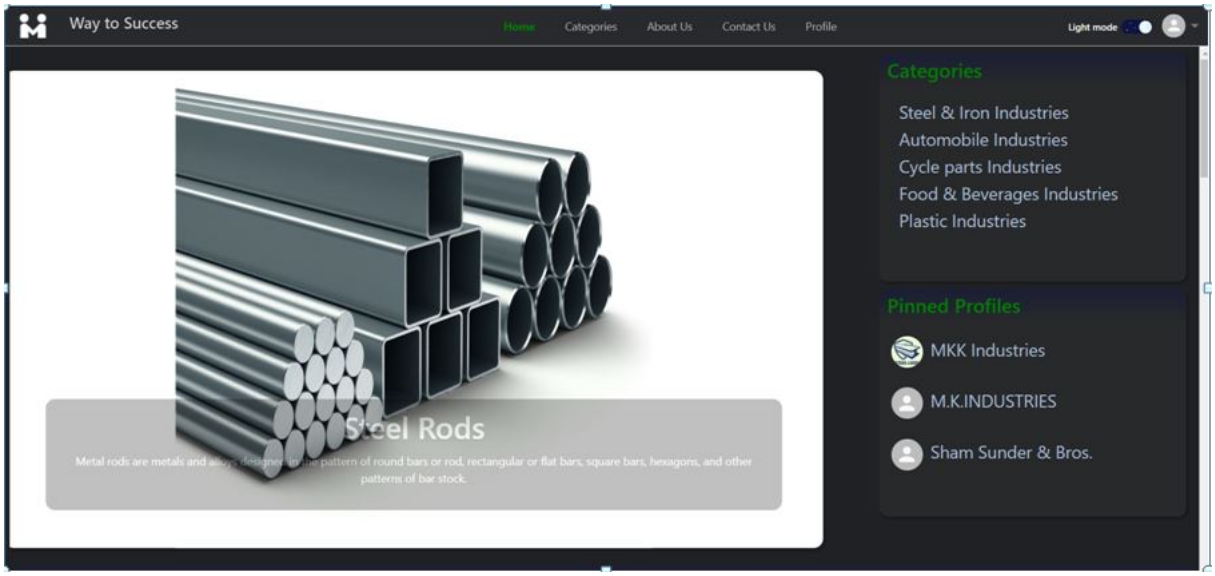


Figure 11: HOME SCREEN 1

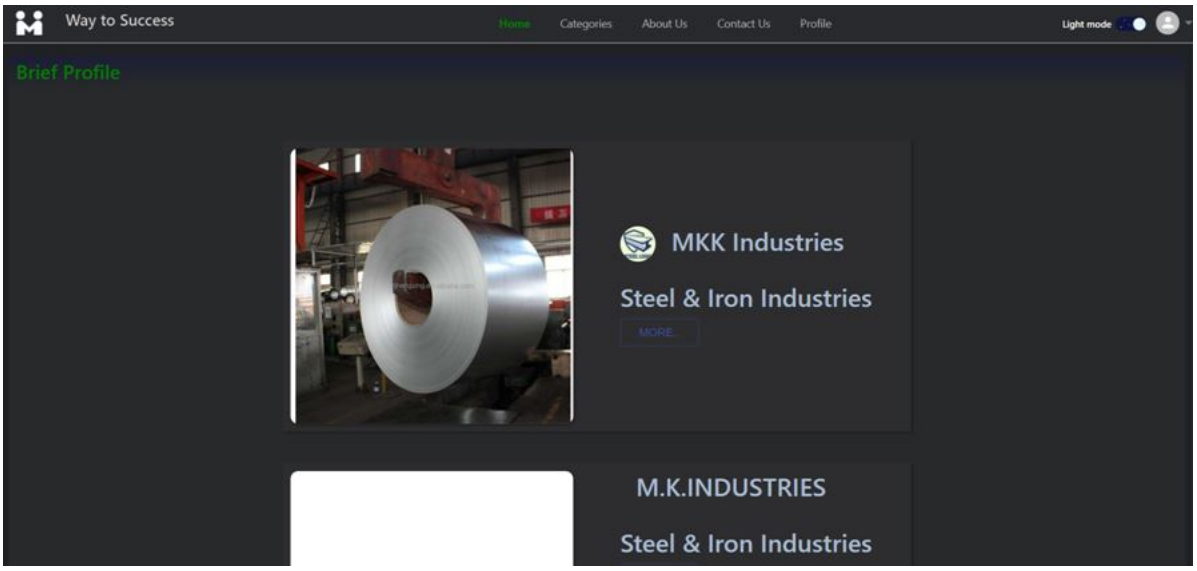


Figure 12: HOME SCREEN 2

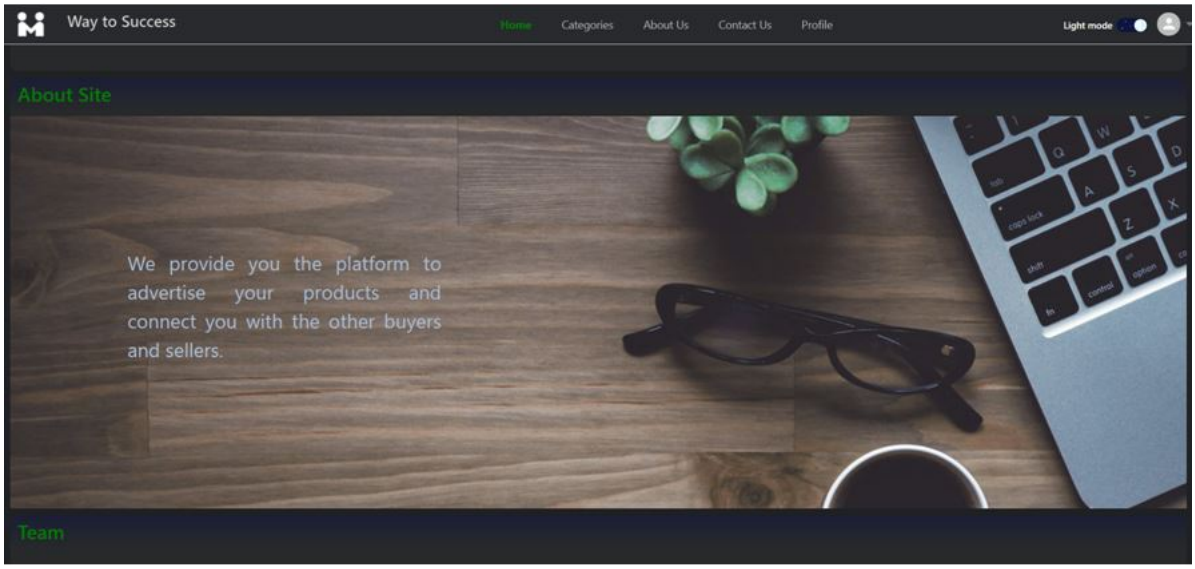


Figure 13: HOME SCREEN 3

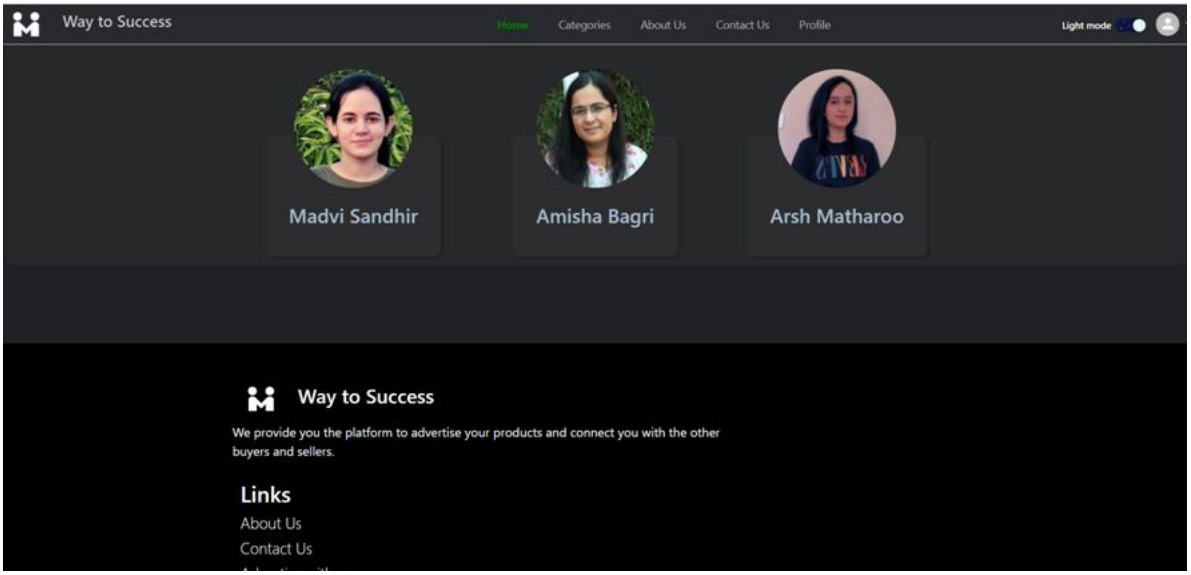


Figure 14: HOME SCREEN 4

LOGIN SCREEN:

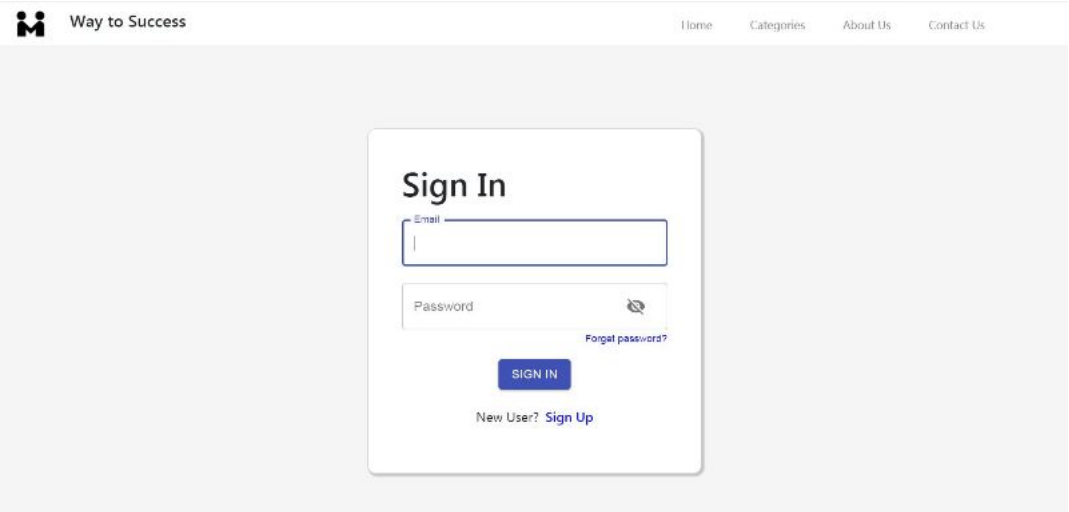


Figure 15: LOGIN SCREEN

REGISTER SCREEN

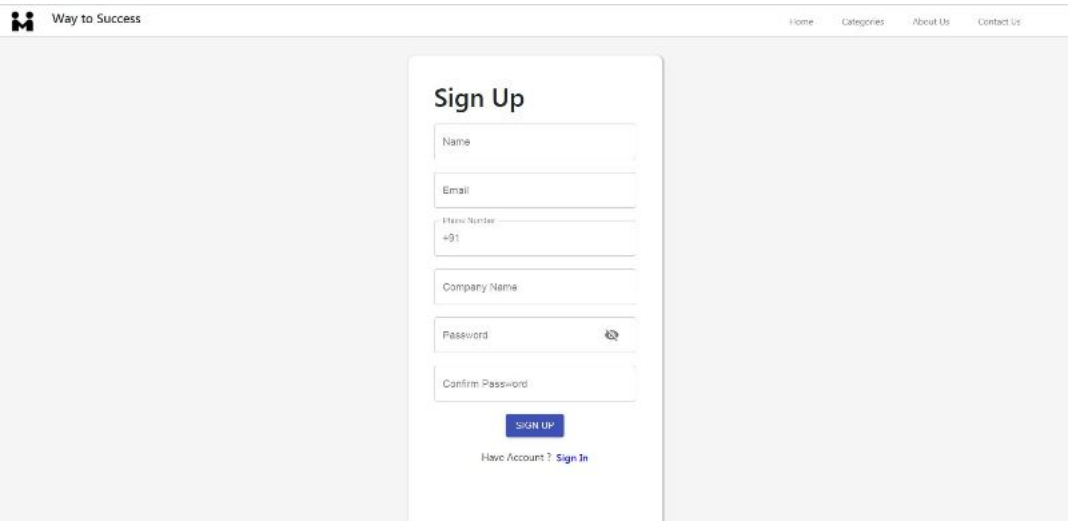


Figure 16: REGISTER SCREEN

CATEGORY SCREEN

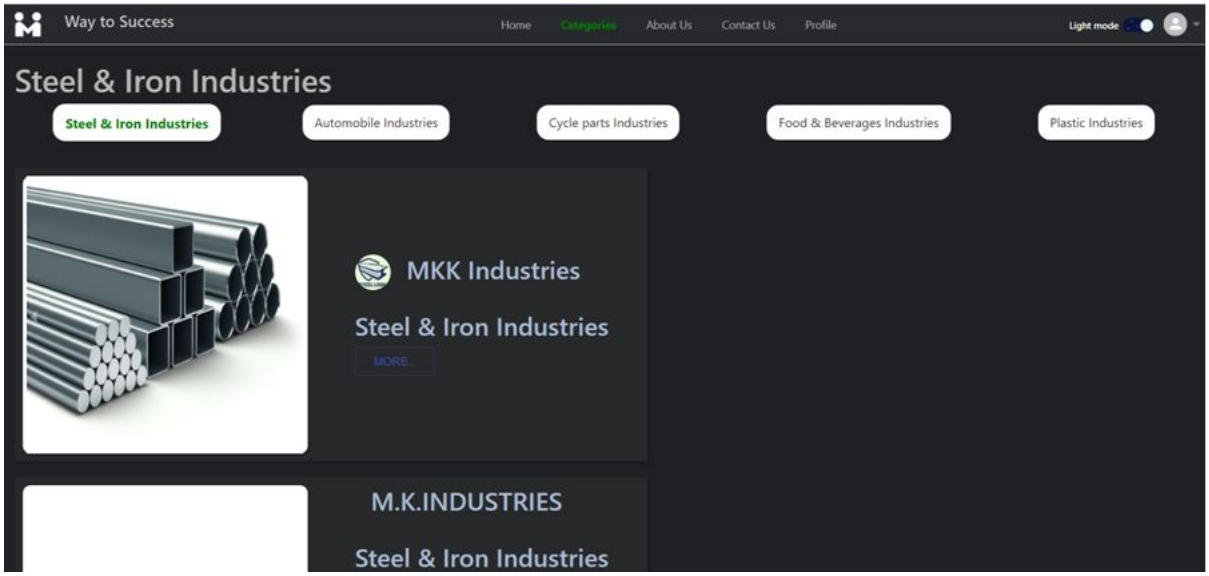


Figure 17: CATEGORY SCREEN

ABOUT US SCREEN

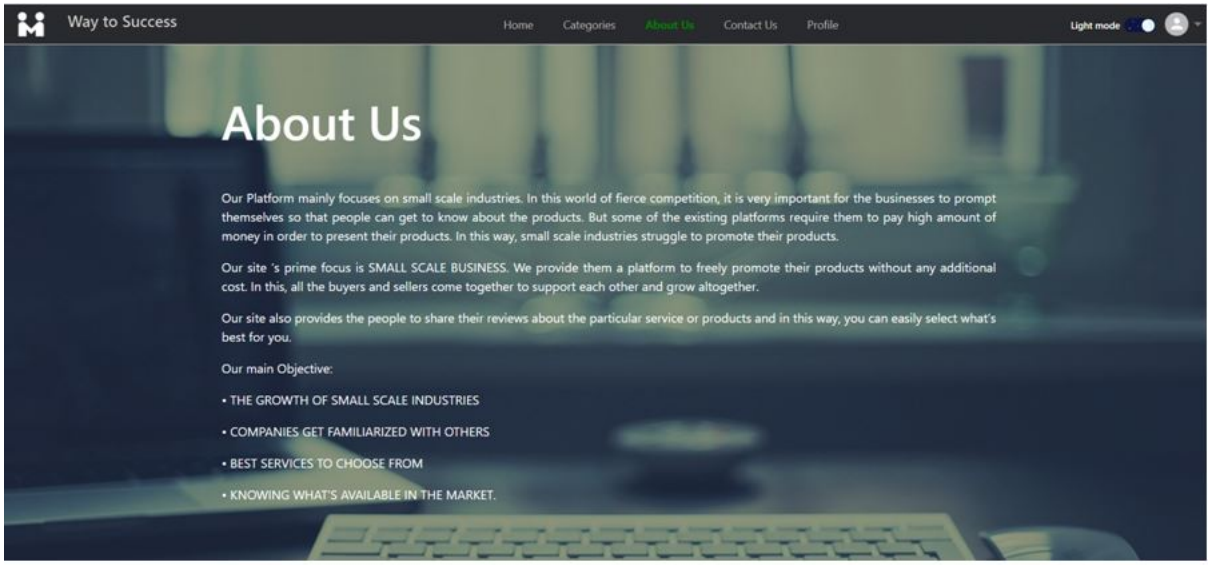


Figure 18: ABOUT SCREEN

CONTACT US

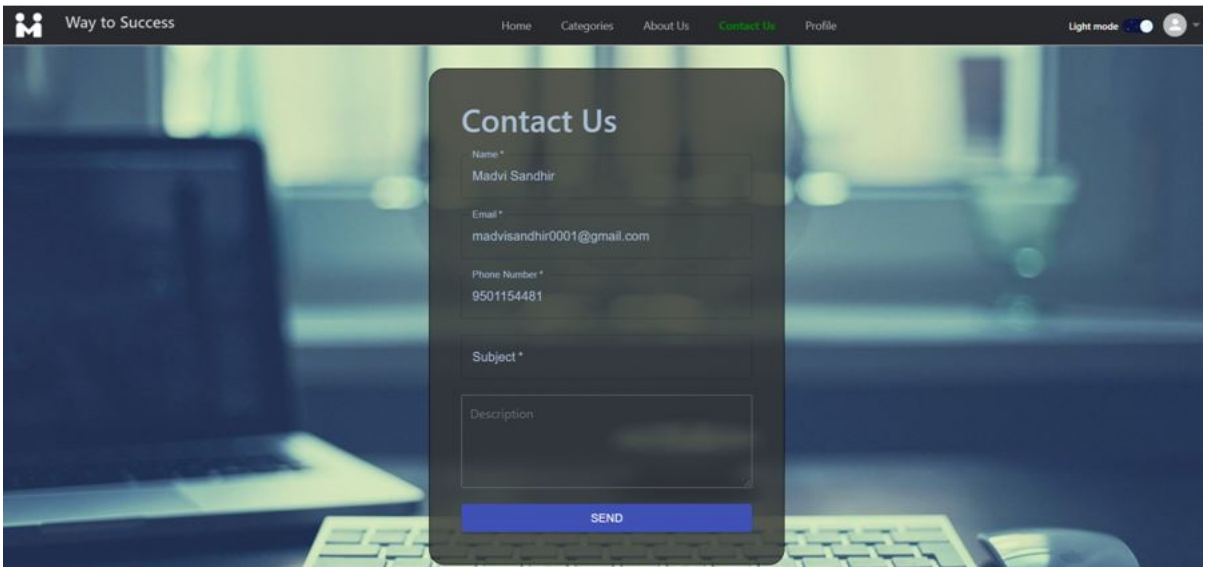


Figure 19: CONTACT US

ADMIN SCREEN

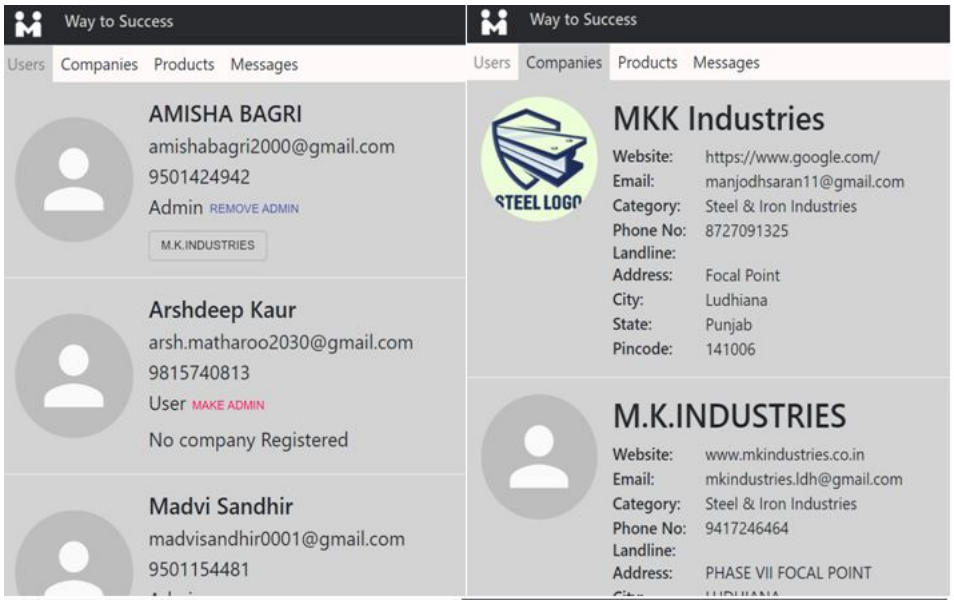


Figure 20: ADMIN SCREEN

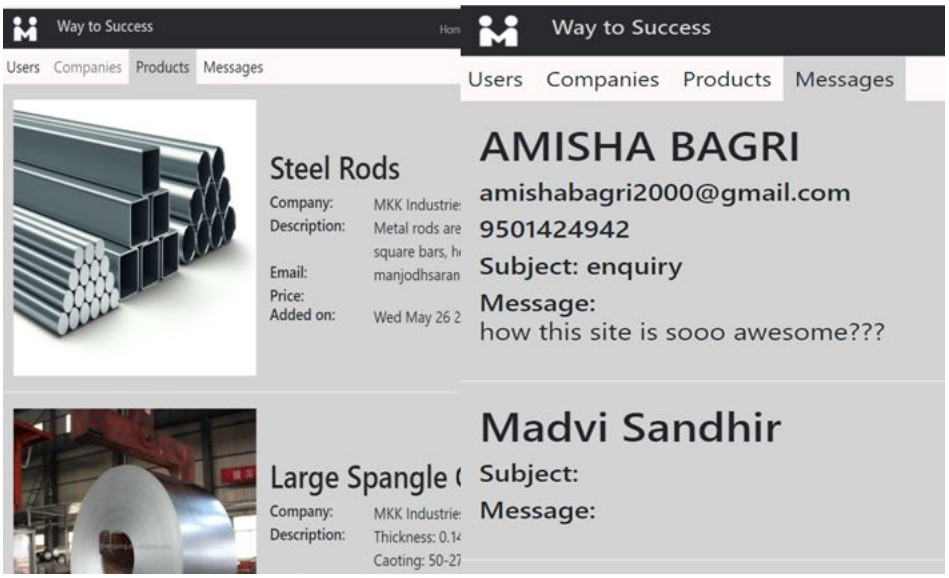


Figure 21: ADMIN SCREEN 2

DASHBOARD SCREEN

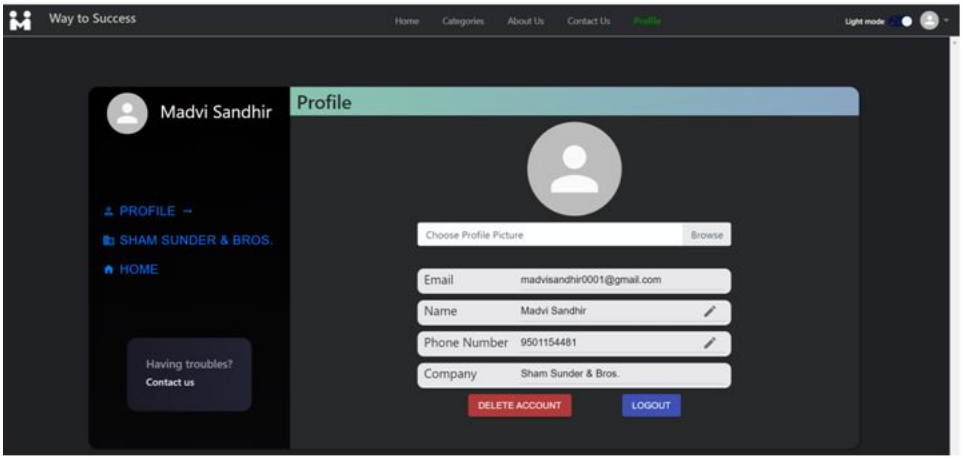


Figure 22: DASHBOARD SCREEN 1

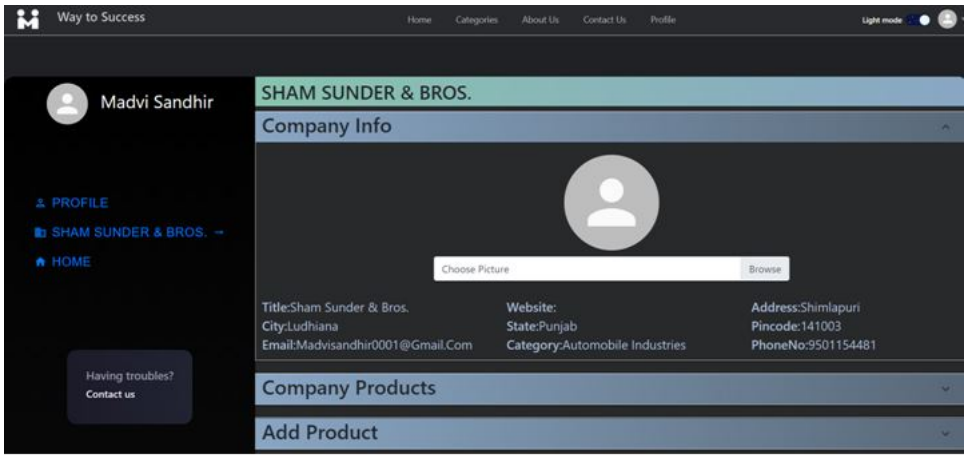


Figure 23: DASHBOARD SCREEN 2

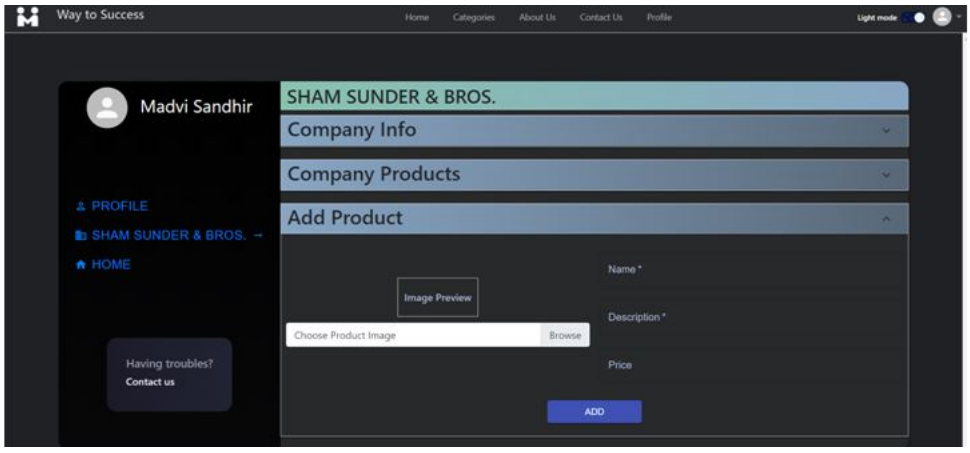


Figure 24: DASHBOARD SCREEN 3

COMPANY SCREEN:

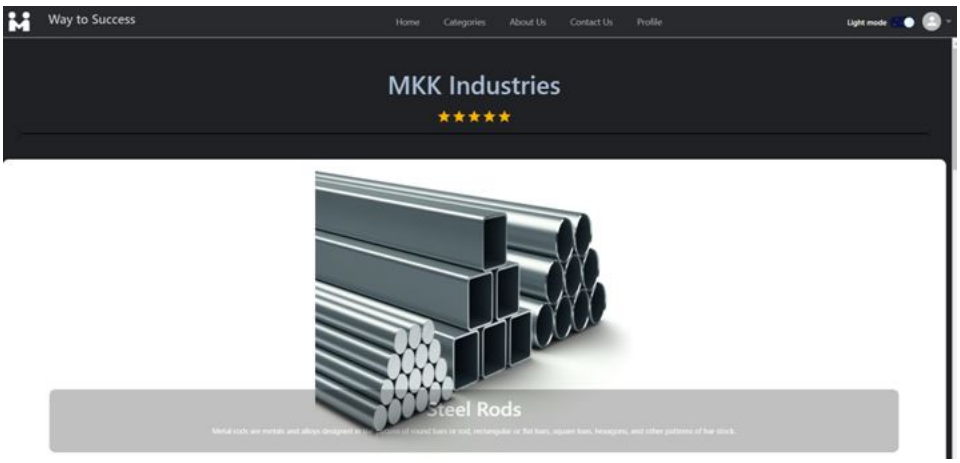


Figure 25: COMPANY SCREEN 1

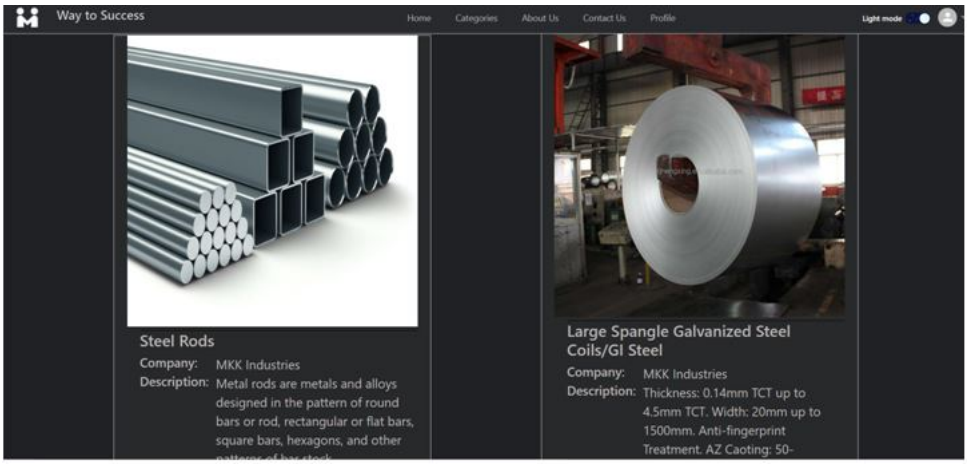


Figure 26: COMPANY SCREEN 2

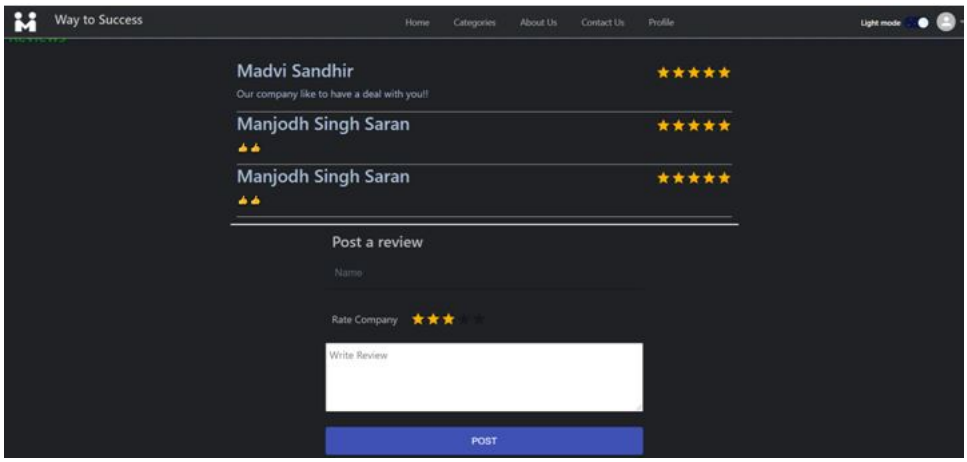


Figure 27: COMPANY SCREEN 3

5.3 Back Ends Representation (Database to be used)

FIREBASE

Firebase is a Backend-as-a-Service — BaaS — that started as a YC11 start-up and grew up into a next generation app-development platform on Google Cloud Platform. Firebase frees developers to focus crafting fantastic user experiences. You don't need to manage servers.

Real-time Database: When you connect your app to Firebase, you're not connecting through normal HTTP. You're connecting through a WebSocket. WebSocket's are much, much faster than HTTP. You don't have to make individual WebSocket calls, because one socket connection is plenty. All of your data syncs automagically through that single WebSocket as fast as your client's network can carry it.

File Storage: Firebase Storage provides a simple way to save binary files — most often images, but it could be anything — to Google Cloud Storage directly from the client. Firebase Storage has its own system of security rules to protect your Google Cloud bucket from the masses, while granting detailed write privileges to your authenticated clients.

Authentication: Firebase authentication has a built-in email/password authentication system. It also supports OAuth2 for Google, Facebook, Twitter and GitHub. We'll focus on email/password authentication for the most part.

Hosting: Firebase includes an easy-to-use hosting service for all of your static files. It serves them from a global CDN (content delivery network) with HTTP/2.

5.3.1 Snapshots of Database Tables with brief description

All Firebase Realtime Database data is stored as JSON objects. You can think of the database as a cloud-hosted JSON tree. Unlike a SQL database, there are no tables or records

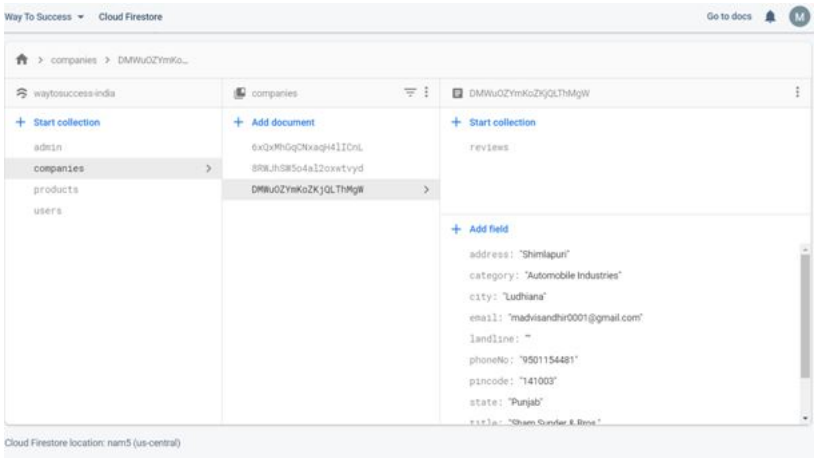


Figure 28: DATABASE SCREEN 1

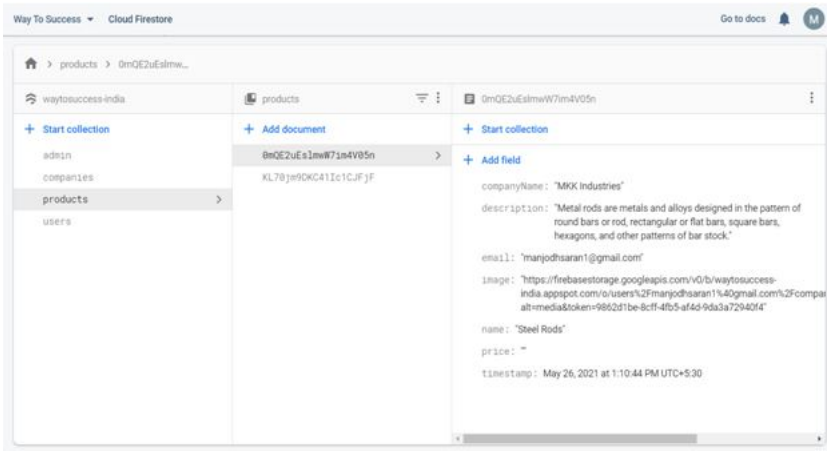


Figure 29: DATABASE SCREEN 2

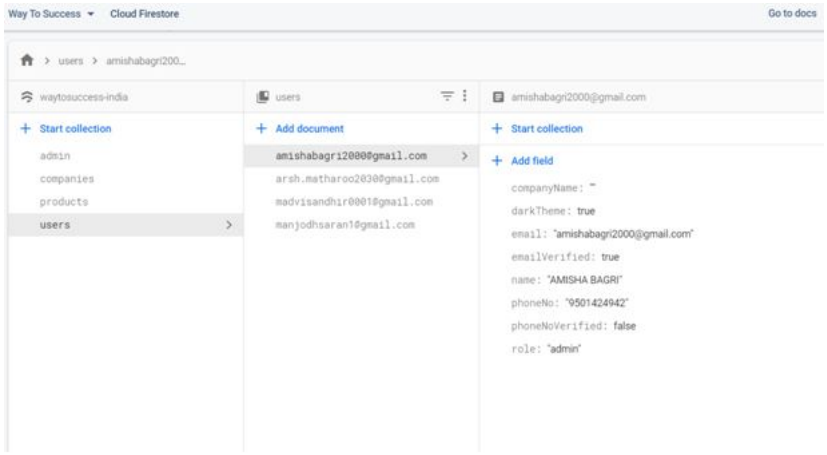


Figure 30: DATABASE SCREEN 3

6 Conclusion and Future Scope

This platform will work perfectly for the Small Scale Industries and will fulfill their needs. This platform will also help them to expand their business and interact with other people and explore a lot more. With the feature to form their own catalouge it will help industries to give a detailed explanation about the services or products they offer which helps to attract more customers. The Rating and review system will also help the users to give their feedback and also help others to choose the best. Overall this platform is betsstor Small Scale Businesses and will definitley help them.

References/Bibliography

Note: It should be in IEEE Format

[1] Nicholas C.Zakas High Performance JavaScript: Build Faster Web Application. [Online]. Available: https://www.oreilly.com/library/view/high-performance-javascript/9781449382308/

[2] Powers,S(2016) Learning Node: Moving to the Server-Side, Second Edition.[Online]. Available:https://oiipdf.com/le node-2nd-edition