

# **INTRODUCTION**

## **OVERVIEW**

This report discusses the result of the work done in development of “Optimizing Courier Delivery System” on JavaScript Platform. The project aims at the development of an application for solving the present difficulties faced by courier delivery agents by optimizing their courier delivery route according to the traffic conditions on the daily basis.

## **BACKGROUND AND MOTIVATION**

Large courier companies use centralized computer systems to delegate delivery jobs to drivers.

If deliverymen carry multiple jobs concurrently these may lead to choose route, which may be sub optimal.

Therefore, with this system we tried to automate the courier delivery system, which can make decision that which delivery point it should reach first for feasible ways by taking list of address from all the nearest delivery points.

Therefore, with our application the system automates the delivery services that can make decisions as to which delivery point should be reached first for feasible ways by taking list of address from all the nearest delivery points.

## **OBJECTIVE**

The final goal of the project is twofold.

1. Integration of google maps into the agent app, which helps the agent from the hassle of entering each and every location before delivering the courier quota for the day.
2. Automation of delivery services so that once the delivery agent enters the address of allotted packages, the system can make decisions as to which

package must be delivered first based on the nearest delivery points, traffic conditions and time taken.

## **METHODOLOGY**

To implement the above goals, the following methodology needs to be followed:

1. Specifying the application and various components of the architecture.
2. Specifying the bindings between the various modules and JavaScript packages.
3. Specifying the server ports between the modules.
4. Analysis: Extracting the required data for analysis and then doing the analysis.

## **ANALYSIS**

On the basis of analysis and literature survey regarding the present difficulties faced by the courier delivery executives, we decided to make this project as our mini project to help facilitate smooth delivery of the packages without any hassle for the delivery executives.

We made this possible by first integrating google maps into our application so that the delivery executive need not open the google map every now and then to check which route needs to be taken to reach sooner to the destination.

Secondly, once the delivery executive pin points all the locations of which he has to deliver the package for the day, our Google API automates the delivery services that can make decisions as to which delivery has to be made first by taking feasible route among all the available routes by taking list of address from all the nearest delivery points.

### **REQUIREMENT ANALYSIS:**

#### **SOFTWARE REQUIREMENTS:**

Operating System: Windows 10 / Ubuntu

Front end: React.js v16.x, Bootstrap v4.2.1, HTML5 & CSS

Back end: Node.js v11.8.0

Database: MongoDB v4.0.1

#### **HARDWARE REQUIREMENTS:**

RAM: 8GB and above

Hard disk: 120GB and above

Processor: Intel i3 and above

## **FUNCTIONAL REQUIREMENTS:**

1. Admin Login and Logout: This module enables the admin to have control over the packages to be delivered and active agents available to deliver the same.
2. Agent Login and Logout: This module enables the agent to know which packages are allotted to them for the day and where it has to be delivered.
3. Integrated Google Maps: Using Google Maps API, we have integrated the same in our application to help the agents deliver the package without any hassle.
4. Automate delivery routes: With this algorithm, we can automate the courier delivery that can make decisions as to which delivery point needs to be reached first by feasible ways from all the nearest delivery points.

## **NON-FUNCTIONAL REQUIREMENTS:**

1. Platform independence: The web application should be responsive on both the popular operating systems Android and iOS.
2. Performance: The application should be able to run on all versions of OS after a limit and should consume less power. The web application should not crash at any condition.
3. Permissions and Authentication: The application should ask for user's permissions to access GPS. No one apart from authorized agent/admin should be able to log in.
4. Live updates: The agent will get live updates when connected to internet.

## **TOOLS AND TECHNOLOGIES:**

### **APPLICATION DEVELOPMENT TECHNOLOGIES:**

This application is built using MERN stack i.e. MongoDB, Express, React.js and Node.js.

MongoDB is an open-source database software which is NoSQL in architecture. It stores data as JSON document. It is fast, reliable and efficient.

Express is a web application framework for Node.js. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.

React.js is a JavaScript library for building user interfaces. Facebook and a community of individual developers and companies maintain it. React can be used as a base in the development of single-page or mobile applications.

## **INTEGRATION TOOLS:**

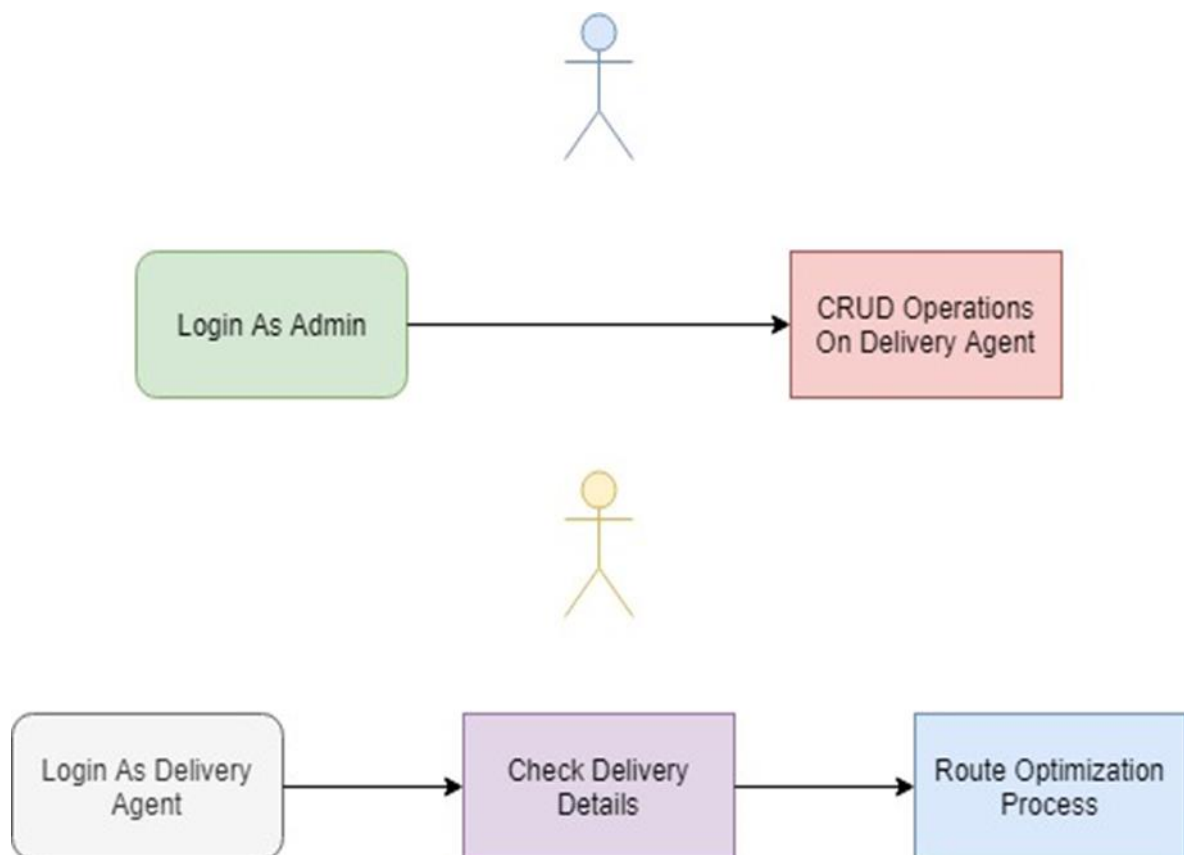
### **GIT AND GITHUB:**

Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity and support for distributed, non-linear workflows.

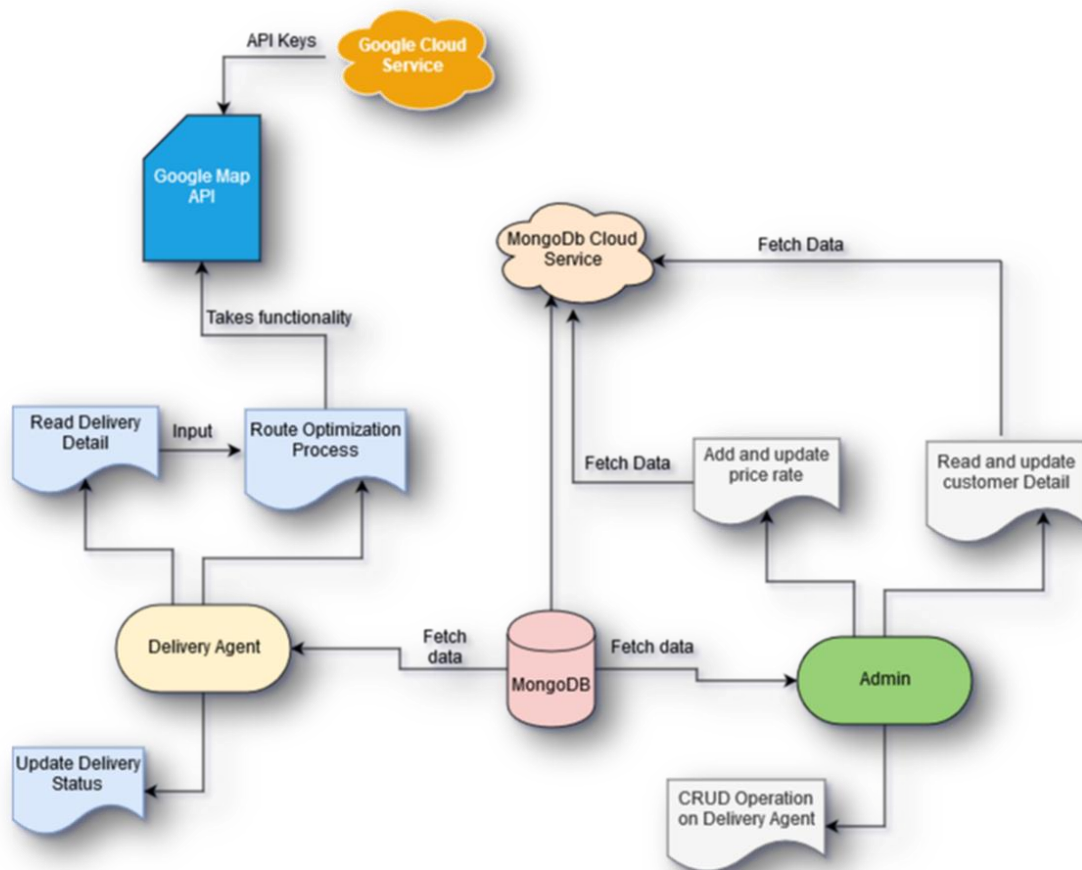
GitHub is a web based hoisting service for version control using Git. It is mostly used for computer code. It offers all of the distributed version control and source code management functionality of Git as adding its own features.

## DESIGN

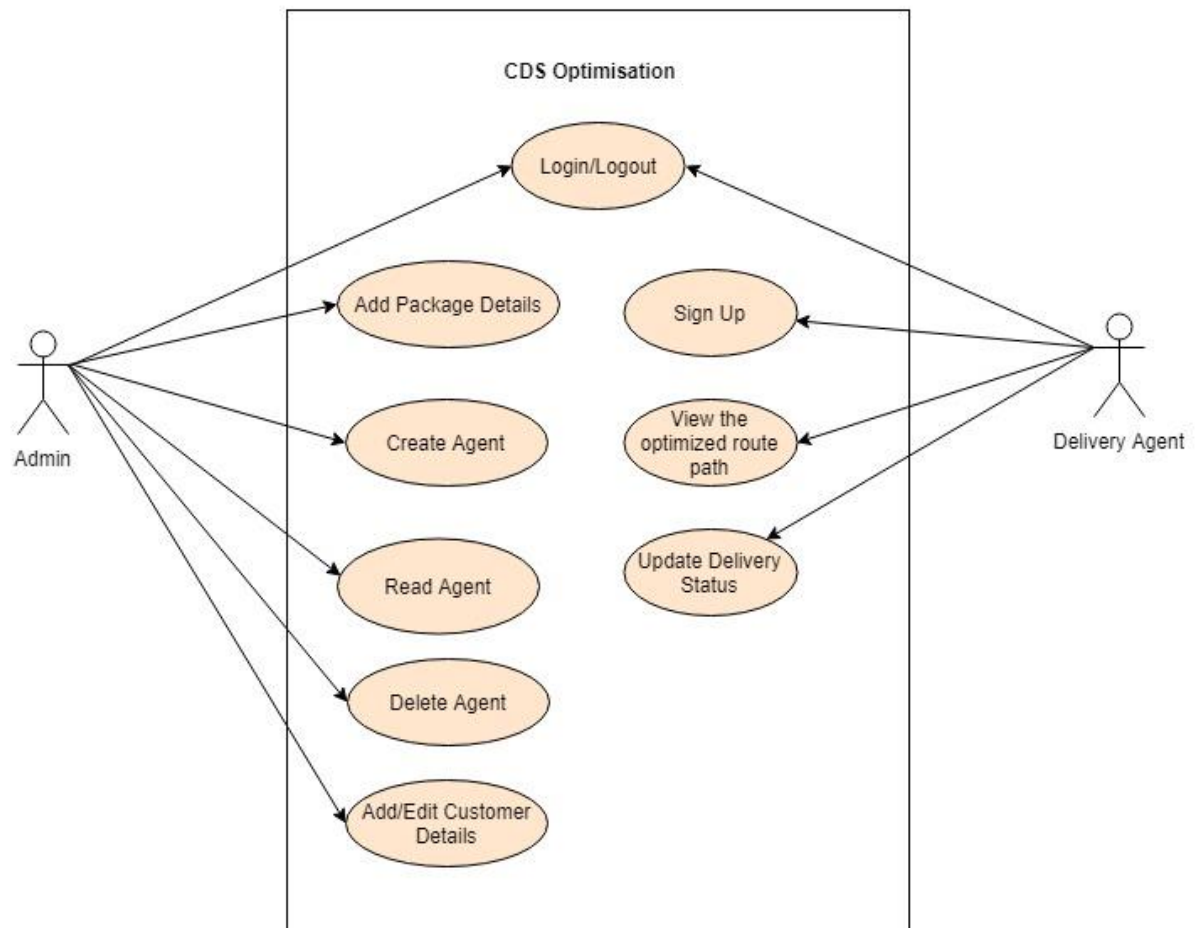
### OVERVIEW OF THE SYSTEM



## ARCHITECTURE OF THE SYSTEM



## USE CASE DIAGRAM



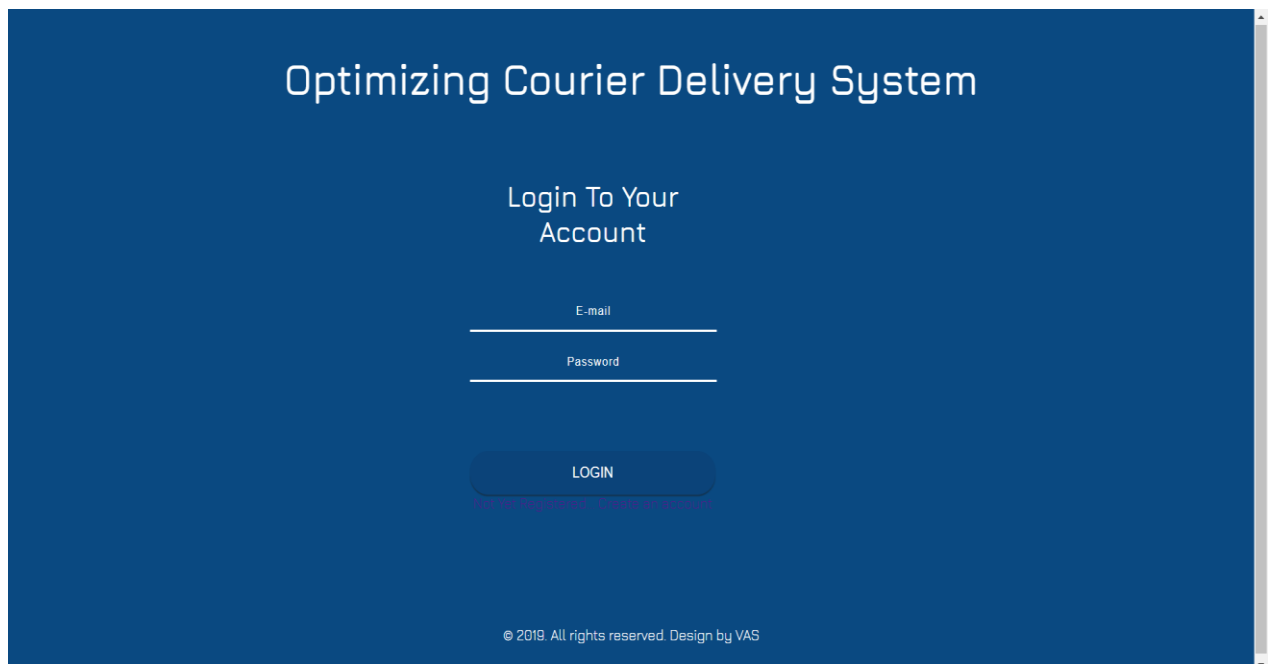
Our project consists of two folds:

1. Admin: Who has control over the application as to which all are the packages to be delivered to the destination and who all are the active agents and which packages are assigned to them for delivery.
2. Agent: Who can see which courier has been allotted to him for the day. Once he starts the map, it automates as to which courier has to be delivered and which feasible route is to be taken based on the traffic conditions and time taken.



## SCREEN SHOTS

### LOGIN SCREEN



The screenshot shows a login interface on a dark blue background. At the top, the title "Optimizing Courier Delivery System" is displayed in white. Below it, the text "Login To Your Account" is centered. There are two input fields: "E-mail" and "Password", each with a white underline. A blue "LOGIN" button is positioned below the fields. A link "Not yet registered? Create an account" is shown in red text below the button. At the bottom, a copyright notice "© 2019. All rights reserved. Design by VAS" is visible.

Optimizing Courier Delivery System

Login To Your Account

E-mail

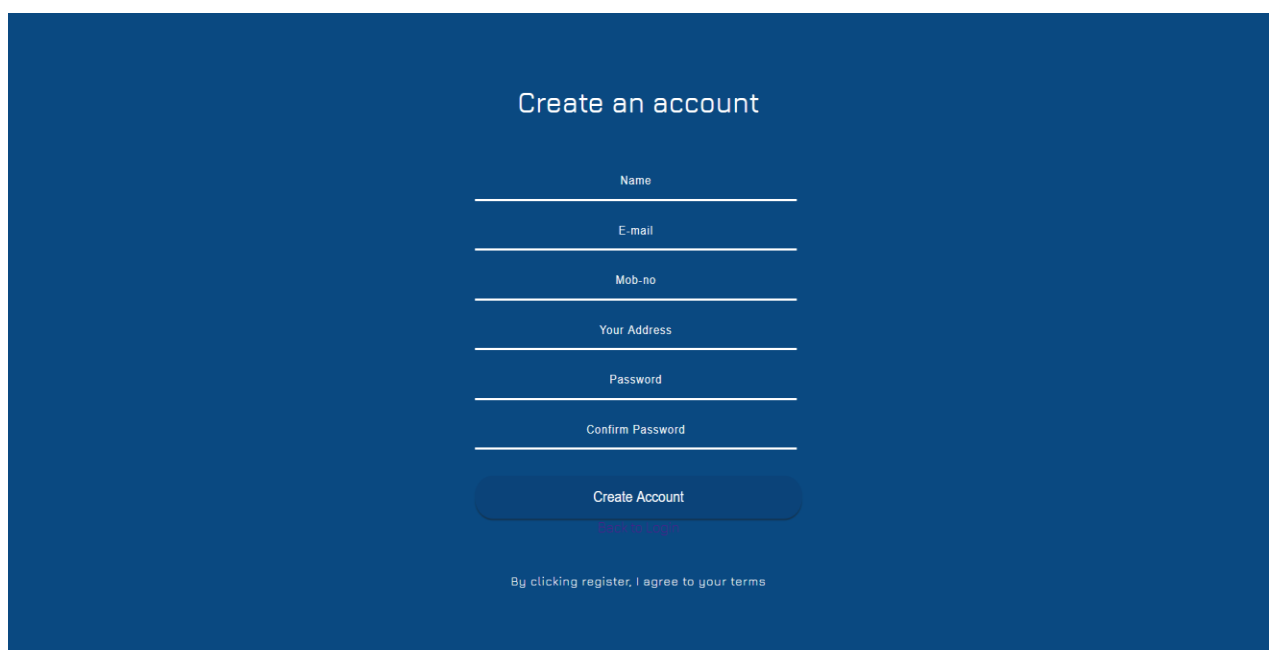
Password

LOGIN

[Not yet registered? Create an account](#)

© 2019. All rights reserved. Design by VAS

### REGISTRATION FORM



The screenshot shows a registration interface on a dark blue background. At the top, the title "Create an account" is displayed in white. Below it, there are six input fields: "Name", "E-mail", "Mob-no", "Your Address", "Password", and "Confirm Password", each with a white underline. A blue "Create Account" button is positioned below the fields. A link "Not yet login?" is shown in red text below the button. At the bottom, a text line reads "By clicking register, I agree to your terms".

Create an account

Name

E-mail

Mob-no

Your Address

Password

Confirm Password

Create Account

[Not yet login?](#)

By clicking register, I agree to your terms

## ADMIN HOMEPAGE

Hello Admin [Home](#) [Create](#) [Index](#) [Agent](#) [Settings](#) [Log Out](#)

### Admin Panel

## PACKAGE CREATION

Hello Admin [Home](#) [Create](#) [Index](#) [Agent](#) [Settings](#) [Log Out](#)

### Admin Panel

#### Add New Package

Person Name:

Package Name:

Package Number:

[Register Parcel](#)

PACKAGE LIST

Hello Admin

[Home](#) [Create](#) [Index](#) [Agent](#) [Settings](#) [Log Out](#)

Admin Panel

Package List

Person	Package	Package Number	Action	
vj	laptop	842342084290	<a href="#">Edit</a>	<a href="#">Delete</a>
Ayush Pratyay	Xiaomi	83948559304	<a href="#">Edit</a>	<a href="#">Delete</a>
Subham Singh	Huawei	48298492348	<a href="#">Edit</a>	<a href="#">Delete</a>

ACTIVE AGENTS LIST

Hello Admin

[Home](#) [Create](#) [Index](#) [Agent](#) [Settings](#) [Log Out](#)

Admin Panel

Active Agents

Agent id	Agent's Name	Package to be delivered		
5c9f6ce33828df43a071e168	vj	laptop	<a href="#">Edit</a>	<a href="#">Delete</a>
5c9f6e7d3828df43a071e169	Ayush Pratyay	Xiaomi	<a href="#">Edit</a>	<a href="#">Delete</a>
5c9f6e8d3828df43a071e16a	Subham Singh	Huawei	<a href="#">Edit</a>	<a href="#">Delete</a>

## DRIVER MODULE

---

↔ Courier Delivery App

**Build** for Courier Route Optimization app with the MERN stack from scratch

This is a Courier Delivery Web App

[REGISTER](#)

[LOG IN](#)

[DRIVER](#)

## DRIVER LOGIN

---

↔ Courier Delivery App

← [BACK TO HOME](#)

**Login** below

Don't have an account? [Register](#)

Email

Password

[LOGIN](#)

## REGISTRATION PAGE

<> Courier Delivery App

← BACK TO HOME

Register below

Already have an account? [Log in](#)

Name

Email

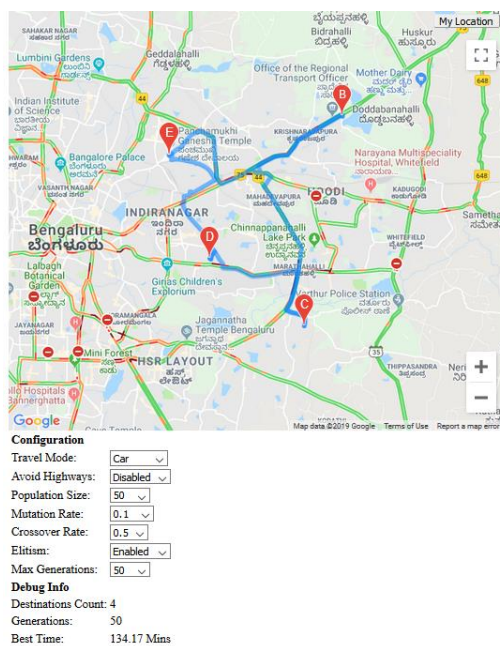
Mobile Number

Password

Confirm Password

Address

## MAP DEMO



## TESTING

<b>Test Case ID</b>	PES_001	<b>Test Case Description</b>	Test the Login Functionality in Admin Panel		
<b>Created By</b>	Vijaykumar R Pai	<b>Reviewed By</b>	Deepthi S Narayan	<b>Version</b>	1
<b>QA Tester's Log</b>					
<b>Tester's Name</b>	Vijaykumar R Pai	<b>Date Tested</b>	30-Mar-2019	<b>Test Case (Pass/Fail/Not Executed)</b>	Pass

S #	Prerequisites:		S #	Test Data
1	Access to Chrome Browser		1	E-mail = vijaykumarrpai@gmail.com
2			2	Password = 12345678
3			3	
4			4	
<b>Test Scenario</b>	Verify on entering valid email and password, the admin can login			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not
1	Navigate to localhost:3000	Admin login page should open	Admin login page is loaded	Pass
2	Enter E-mail & Password	Credentials can be entered	Credentials are entered	Pass
3	Click Login	Admin is logged in	Admin logged in successfully	Pass

<b>Test Case ID</b>	PES_002	<b>Test Case Description</b>	Test the Registration form Functionality in Admin Panel		
<b>Created By</b>	Vijaykumar R Pai	<b>Reviewed By</b>	Deepthi S Narayan	<b>Version</b>	1
<b>QA Tester's Log</b>					
<b>Tester's Name</b>	Vijaykumar R Pai	<b>Date Tested</b>	30-Mar-2019	<b>Test Case (Pass/Fail/Not Executed)</b>	Pass

S #	Prerequisites:		S #	Test Data
1	Access to Chrome Browser		1	Name = Vijaykumar R Pai
2			2	E-mail = <a href="mailto:vijaykumarrpai@gmail.com">vijaykumarrpai@gmail.com</a>
3			3	Mob-no = 7582455420
4			4	Your Address = Blore
			5	Password = 12345678
			6	Confirm Password = 12345678
<b>Test Scenario</b>	Verify on entering name, email, mob-no, address, password and confirm password, the person can register as admin			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not
1	Navigate to localhost:3000	Admin login page should open and navigate to registration form	Admin page loaded and on click navigated to registration page	Pass
2	Enter name, email, mob-no, address, password and confirm password	Credentials can be entered	Credentials are entered	Pass
3	Click Create account	Admin account is created	Account created successfully	Pass

<b>Test Case ID</b>	PES_003	<b>Test Case Description</b>	Test adding new package Functionality in Admin Panel		
<b>Created By</b>	Vijaykumar R Pai	<b>Reviewed By</b>	Deepthi S Narayan	<b>Version</b>	1
<b>QA Tester's Log</b>					
<b>Tester's Name</b>	Vijaykumar R Pai	<b>Date Tested</b>	30-Mar-2019	<b>Test Case (Pass/Fail/Not Executed)</b>	Pass

S #	Prerequisites:		S #	Test Data
1	Access to Chrome Browser		1	Person name = vj
2			2	Package name = Laptop
3			3	Package number = 5845098591
<b>Test Scenario</b>	Verify on entering name, email, mob-no, address, password and confirm password, the person can register as admin			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not
1	Navigate to localhost:3000	Admin login page should open	Admin login page loaded	Pass
2	Enter E-mail & Password	Credentials can be entered	Credentials are entered	Pass
3	Click Login	Admin is logged in	Admin logged in successfully	Pass
4	Navigate to create section	Page is loaded	Toggled to create section	Pass
5	Enter person name, package name and package number	Data is entered	Required fields are entered	Pass
6	Click Register Parcel	Data is stored in DB	Data stored successfully into DB	Pass



<b>Test Case ID</b>	PES_004	<b>Test Case Description</b>	Test adding Package list Functionality in Admin Panel		
<b>Created By</b>	Vijaykumar R Pai	<b>Reviewed By</b>	Deepthi S Narayan	<b>Version</b>	1
<b>QA Tester's Log</b>					
<b>Tester's Name</b>	Vijaykumar R Pai	<b>Date Tested</b>	30-Mar-2019	<b>Test Case (Pass/Fail/Not Executed)</b>	Pass

S #	Prerequisites:		S #	Test Data
1	Access to Chrome Browser		1	Person name = vj
2			2	Package name = Laptop
3			3	Package number = 5845098591
<b>Test Scenario</b>	Verify on clicking the delete button in package list, the respective package is deleted.			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not
1	Navigate to localhost:3000	Admin login page should open	Admin login page loaded	Pass
2	Enter E-mail & Password	Credentials can be entered	Credentials are entered	Pass
3	Click Login	Admin is logged in	Admin logged in successfully	Pass
4	Navigate to index section	Page is loaded	Page loaded successfully	Pass
5	Click delete to remove a particular package	Package is deleted	Package deleted successfully	Pass

<b>Test Case ID</b>	PES_005	<b>Test Case Description</b>	Test Agent list Functionality in Admin Panel		
<b>Created By</b>	Vijaykumar R Pai	<b>Reviewed By</b>	Deepthi S Narayan	<b>Version</b>	1
<b>QA Tester's Log</b>					
<b>Tester's Name</b>	Vijaykumar R Pai	<b>Date Tested</b>	30-Mar-2019	<b>Test Case (Pass/Fail/Not Executed)</b>	Pass

S #	Prerequisites:		S #	Test Data
1	Access to Chrome Browser		1	Person name = vj
2			2	Package name = Laptop
3			3	Package number = 5845098591
<b>Test Scenario</b>	Verify on clicking the delete button in package list, the respective package is deleted.			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not
1	Navigate to localhost:3000	Admin login page should open	Admin login page loaded	Pass
2	Enter E-mail & Password	Credentials can be entered	Credentials are entered	Pass
3	Click Login	Admin is logged in	Admin logged in successfully	Pass
4	Navigate to agent section	Page is loaded	Page loaded successfully	Pass
5	Click delete to remove a particular agent	Agent is deleted	Agent deleted from DB successfully	Pass

Test Case ID		PES_006	Test Case Description		Test the Login Functionality in Agent Panel					
Created By		Ayush Pratyay	Reviewed By		Deepthi S Narayan		Version		2.1	
QA Tester's Log										
Tester's Name		Ayush Pratyay	Date Tested		31-03-2019		Test Case (Pass/Fail/Not Executed)		Pass	
S #	Prerequisites:				S #	Test Data				
1	Access to Chrome Browser				1	E-mail id = Ayush@gmail.com				
2					2	Pass = arpit1				
3					3					
4					4					
Test Scenario		Verify on entering valid user id and password, the Agent can login								
Step #	Step Details		Expected Results		Actual Results			Pass / Fail / Not executed /		
1	Navigate to <a href="http://localhost:3000/">http://localhost:3000/</a>		Agent login page should open		Agent login page loaded			Pass		
2	Enter E-mail id & Password		Credentials can be entered		Credentials are entered			Pass		
3	Click Login		Agent is logged in		Agent logged in successfully			Pass		

Test Case ID		PES_007	Test Case Description		Test the Registration Functionality in Agent Panel				
Created By		Ayush Pratyay	Reviewed By		Deepthi S Narayan		Version		2.1
QA Tester's Log									
Tester's Name		Ayush Pratyay	Date Tested		31-03-2019		Test Case (Pass/Fail/Not Executed)		Pass
S #	Prerequisites:			S #	Test Data				
1	Access to Chrome Browser			1	Name = Ayush				
2				2	Email Id = Ayush@gmail.com				
3				3	Mob-no = 9876543210				
4				4	Address = Banshankari				
5				5	Password = arpit1				
6				6	Conf password = arpit1				
Test Scenario		Verify on entering name, email, mob-no, address, password and confirm password, the person can register as agent							
Step #	Step Details		Expected Results		Actual Results			Pass / Fail / Not executed /	
1	Navigate to <a href="http://localhost:3000/">http://localhost:3000/</a>		Agent registration page should open and navigate		Agent login page loaded			Pass	
2	Enter name, email, mob-no, address, password and confirm password		Credentials can be entered		Credentials are entered			Pass	
3	Click Register		Agent account is created		Agent account created successfully			Pass	

Test Case ID		PES_008	Test Case Description		Test the Login Functionality in Driver module			
Created By		Ayush Pratyay	Reviewed By		Deepthi S Narayan		Version	2.1
QA Tester's Log								
Tester's Name		Ayush Pratyay	Date Tested		31-03-2019		Test Case (Pass/Fail/Not Executed)	Pass
S #	Prerequisites:			S #	Test Data			
1	Access to Chrome Browser			1	E-mail id = Ayush@gmail.com			
2				2	Pass = arpit1			
3				3				
4				4				
Test Scenario	Verify on entering valid userid and password, the Driver can login							
Step #	Step Details		Expected Results		Actual Results		Pass / Fail / Not executed /	
1	Navigate to <a href="http://localhost:3000/">http://localhost:3000/</a>		Driver login page should open		Driver login page loaded		Pass	
2	Enter E-mail id & Password		Credentials can be entered		Credentials are entered		Pass	
3	Click Login		Driver is logged in		Driver logged in successfully		Pass	

## **CONCLUSION**

The objective of the project was to solve the difficulties faced by the courier delivery agents as to which courier has to be delivered first, which has been solved with our application as it shows feasible route once the delivery agent enters the location of the package to be delivered based on traffic conditions and time taken. Since our application is built with React JS, it is lightweight, responsive, loads faster once it will be deployed. Hence, the objective has been fulfilled and application shows optimized route for delivery of the courier.

## BIBLIOGRAPHY

- 1) <https://medium.freecodecamp.org/the-react-handbook-b71c27b0a795>
- 2) <https://medium.com/javascript-in-plain-english/full-stack-mongodb-react-node-js-express-js-in-one-simple-app-6cc8ed6de274>
- 3) <https://appdividend.com/category/react-js/>
- 4) <https://www.npmjs.com/package/node-sms-send>
- 5) <https://stackoverflow.com/questions/42444909/which-is-the-best-place-to-learn-react-js>
- 6) Software Engineering – A Practitioner’s Approach, Roger S Pressman
- 7) React.js Essentials – Artemij Fedosejev