**INTRODUCTION**

**OVERVIEW**

This report discusses the result of the work done in development of “Bulk Email Aggregator” on JavaScript Platform. The project aims at the development of an application to enable Customer Relationship Managers of a company to send bulk emails for collecting feedback of their own electronic products.

**BACKGROUND AND MOTIVATION**

Electronic product companies assign the task of collecting feedback to the CRM’s.

Companies like Croma send text messages to customers to collect feedback where they have to send individual messages to each person.

Therefore, with this system we are trying to send emails to collect feedback from the customers who had purchased products from their company much more efficiently and getting abstract of performance of the electronics in the market.

**OBJECTIVE**

The final goal of the project is twofold.

1. A web application that is used to provide a paid service to the end user which will enable them to send bulk emails for collecting feedback of their own electronic products once they do payment through the Stripe payment gateway which we have used in the application.
2. Since our service will be provided on usage of assigned credit points to user. So once the Customer Relationship Manager (CRM) who will be using the application does transaction, a credit of 5 points will be automatically added to the individual’s account and then that person can make the best use of the points by using it to send bulk email to as many people as needed. Each time the person sends a bulk email, 1 credit point is deduced.

**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 CRM’s of companies like Croma as they have to send individual emails or text messages to collect feedback from each person.

Therefore, with the system of ours we are trying to send bulk emails to collect feedback from the customers who has purchased products from their company much more efficiently and getting abstract of performance of the electronics in the market.

Secondly, our application works based on assigned credit points to user. Therefore, once the CRM does transaction through our app, 5 credit points will be automatically added to the account and the user can make the best use of the credit points to send bulk email as 1 credit point get deduced each time a mail is sent.

**REQUIREMENT ANALYSIS**:

**SOFTWARE REQUIREMENTS**:

Operating System: Windows 10 / Ubuntu

Front end: React.js v16.x

Back end: Node.js v11.x

Database: MongoDB Atlas

Authentication API: Google OAuth

Payment Gateway API: Stripe

Cloud Deployment: Heroku Deployment

Other Technologies used: Twilio SendGrid, Git and GitHub

**HARDWARE REQUIREMENTS:**

RAM: 8GB and above

Hard disk: 120GB and above

Processor: Intel i3 and above

**FUNCTIONAL REQUIREMENTS**:

1. CRM Signs Up via **Google OAuth.**
2. CRM **Pays for email credits** via Stripe.
3. CRM **creates a new feedback.**
4. CRM **enters list of emails to collect feedback.**
5. Application **send emails to list of customers.**
6. **Customers click on the link provided in the email to send feedback.**
7. Application will **tabulate feedback received**.
8. CRM **can see list of all feedback responses.**
9. Logout

**NON-FUNCTIONAL REQUIREMENTS**:

1. Availability
2. Maintainability
3. Performance
4. Supportability

**TOOLS AND TECHNOLOGIES**:

**APPLICATION DEVELOPMENT TECHNOLOGIES**:

This application is built using MERN stack ie MongoDB (MongoDB Atlas), 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.

**Node.js®** is a JavaScript runtime built on [Chrome's V8 JavaScript engine](https://v8.dev/).

**Google OAuth** use the [OAuth 2.0 protocol](http://tools.ietf.org/html/rfc6749) for authentication and authorization. Google supports common OAuth 2.0 scenarios such as those for web server, installed, and client-side applications.

**Stripe** is a service that allows users to accept payments online, specifically developers. With the **Stripe** application, users can keep track of payments, search past payments, create recurring charges, and keep track of customers.

**Twilio SendGrid** provides a [cloud-based](https://en.wikipedia.org/wiki/Cloud_computing) service that assists businesses with email delivery.

**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.

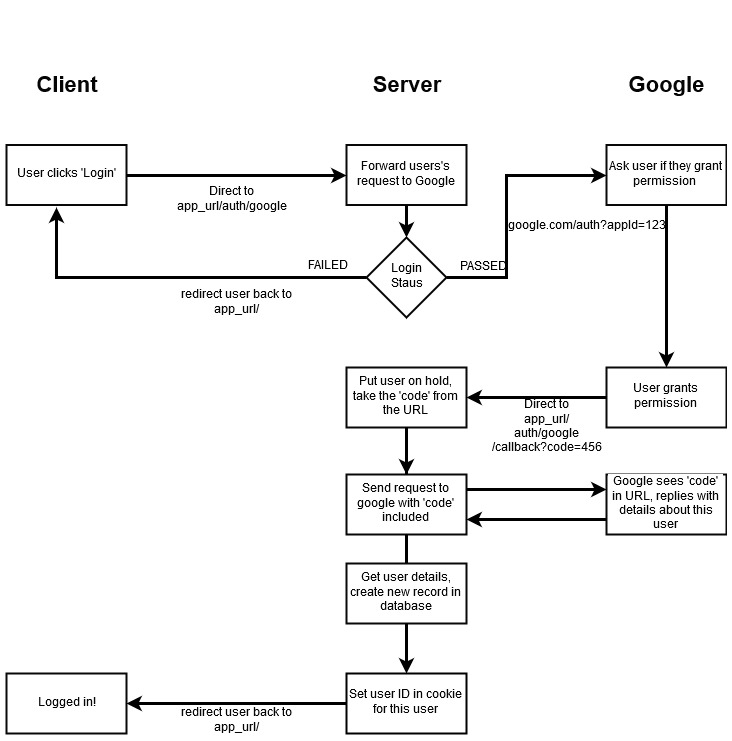
GitHub is a web based hoisting service for version control using Git. It is mostly used for computer code. It offers all of the distributed version control and source code management functionality of Git as adding its own features.

**HEROKU**:

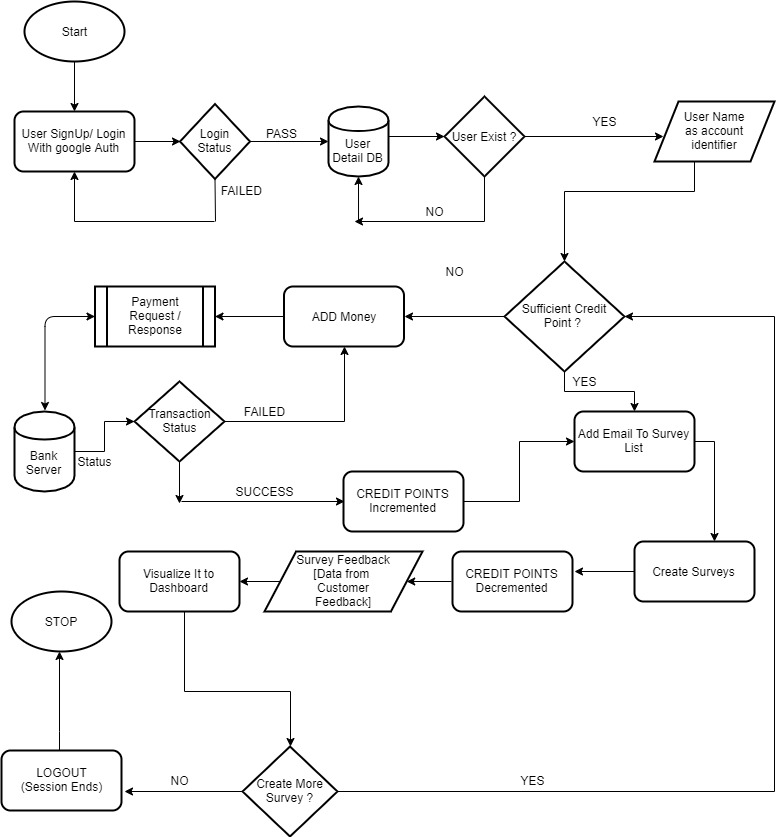
**Heroku** is a container-based cloud Platform as a Service (PaaS). Developers use **Heroku** to deploy, manage, and scale modern apps. Our platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market.

**DESIGN**

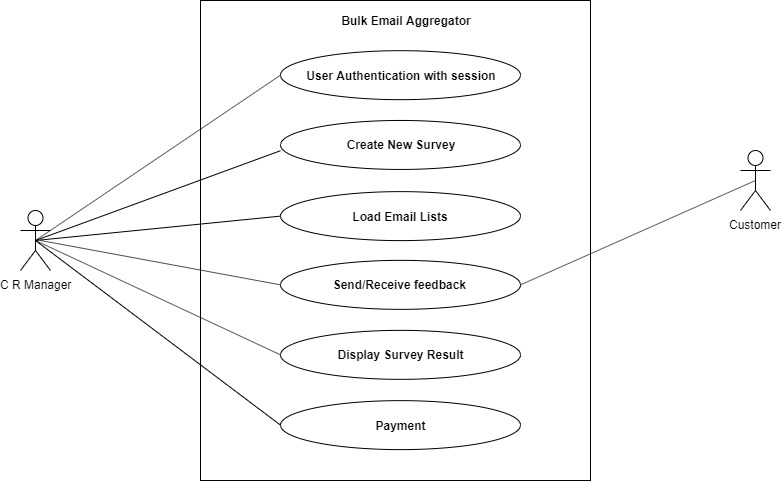
**GOOGLE OAUTH FLOW DIAGRAM**



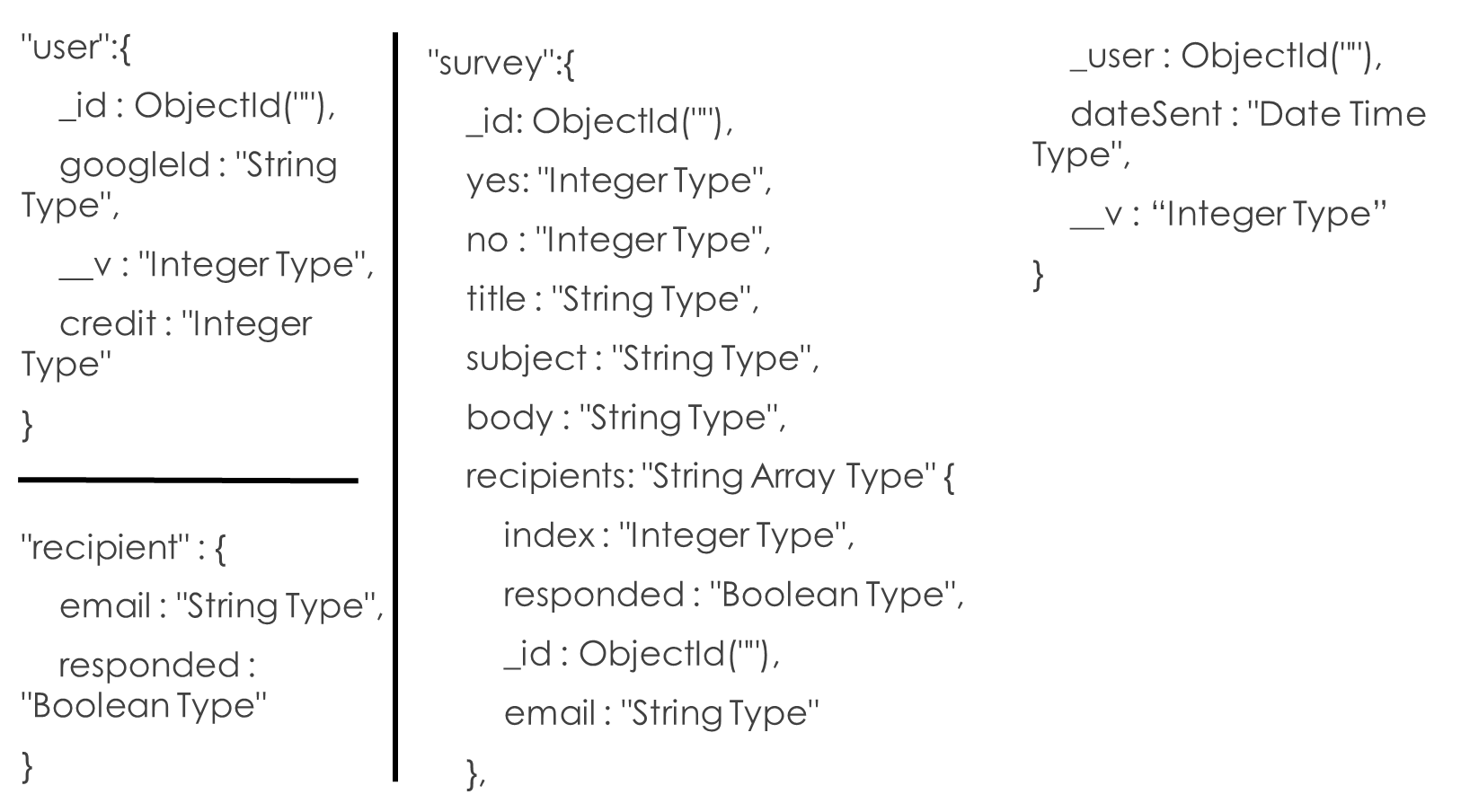
**FLOW DIAGRAM OF THE SYSTEM**



**USE CASE DIAGRAM**



**DOCUMENT STRUCTURE**

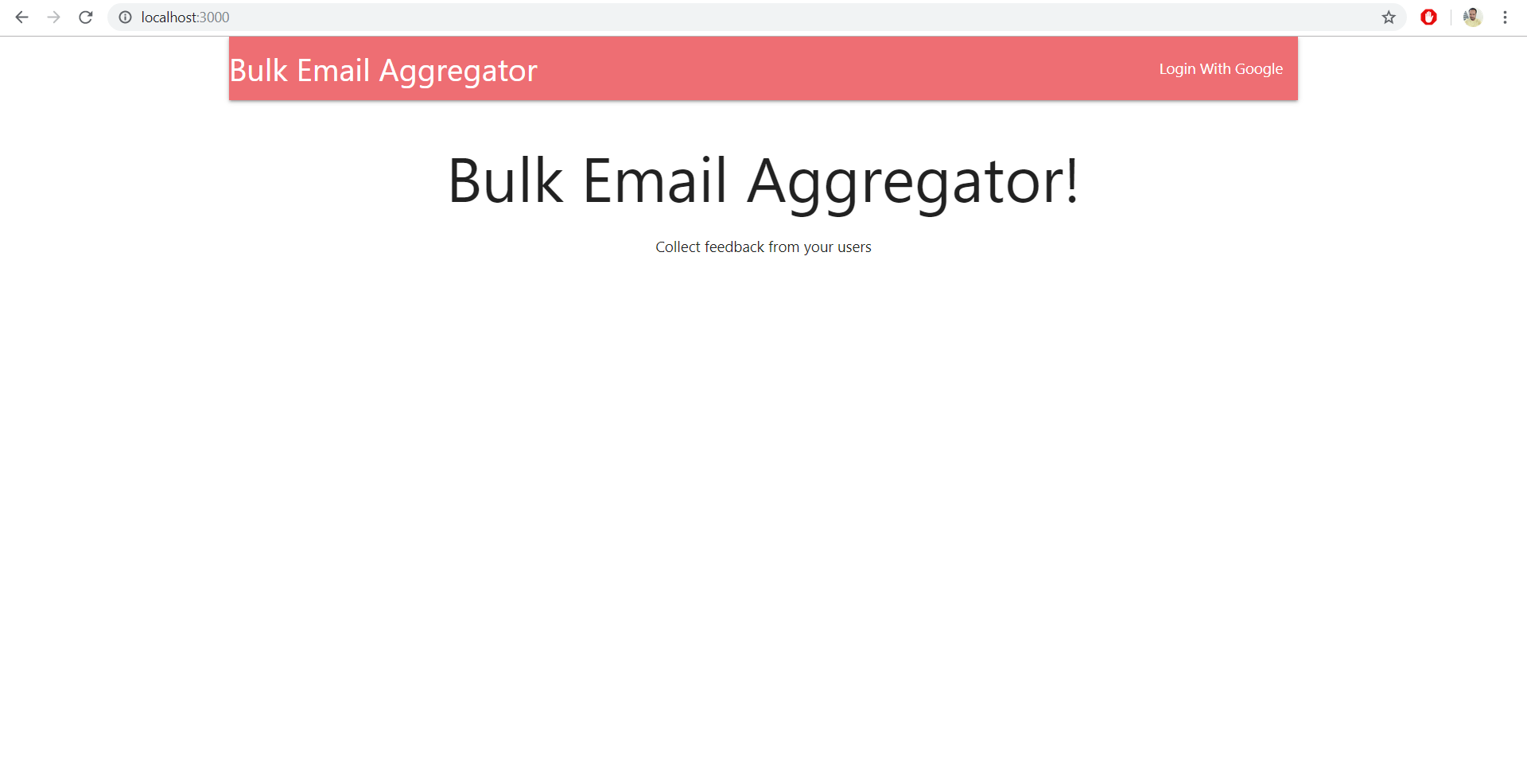
****

Our project consists of two folds:

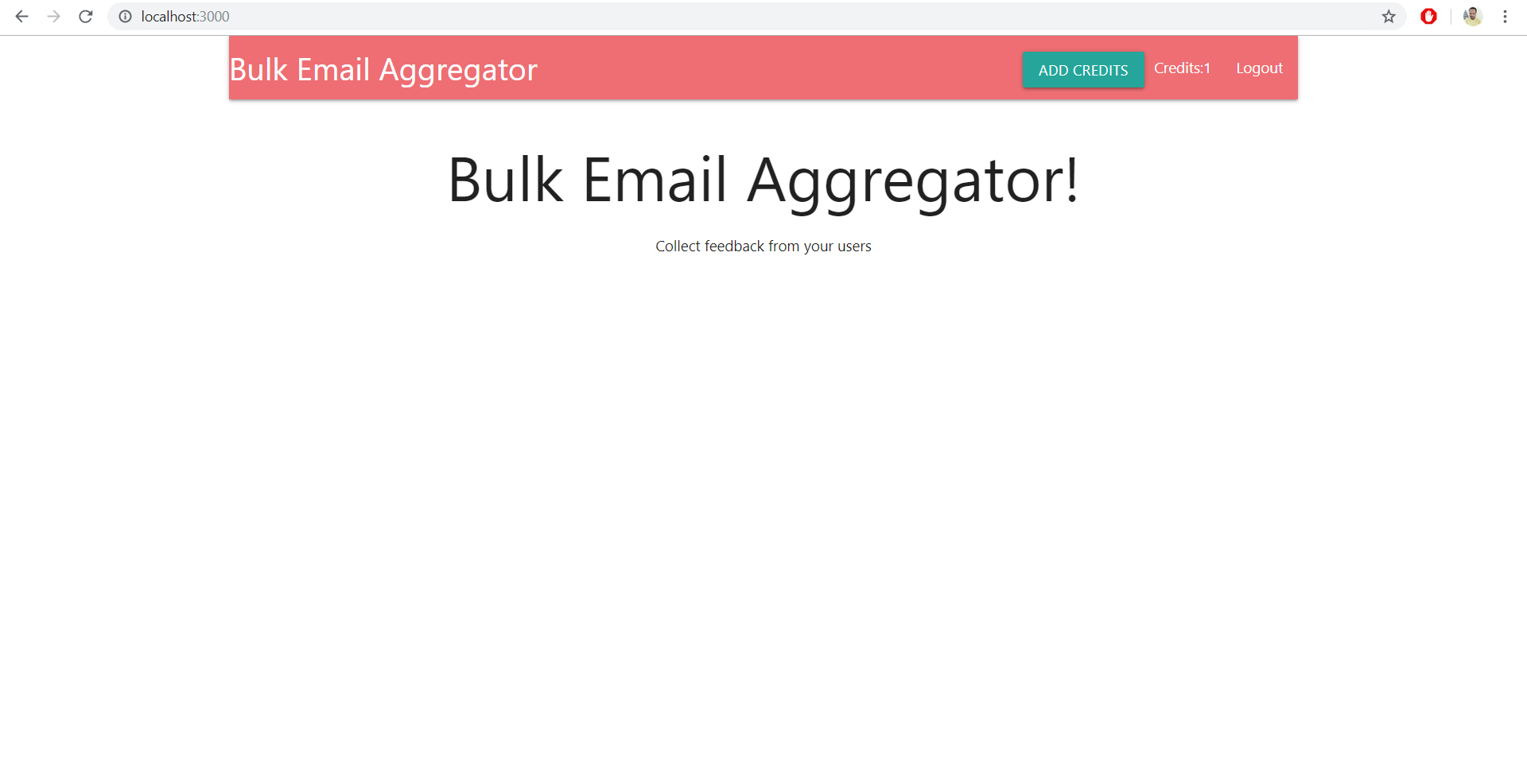
1. Customer Relationship Manager (CRM).
2. Customers: Who purchase products from different electronic companies.

**SCREEN SHOTS**

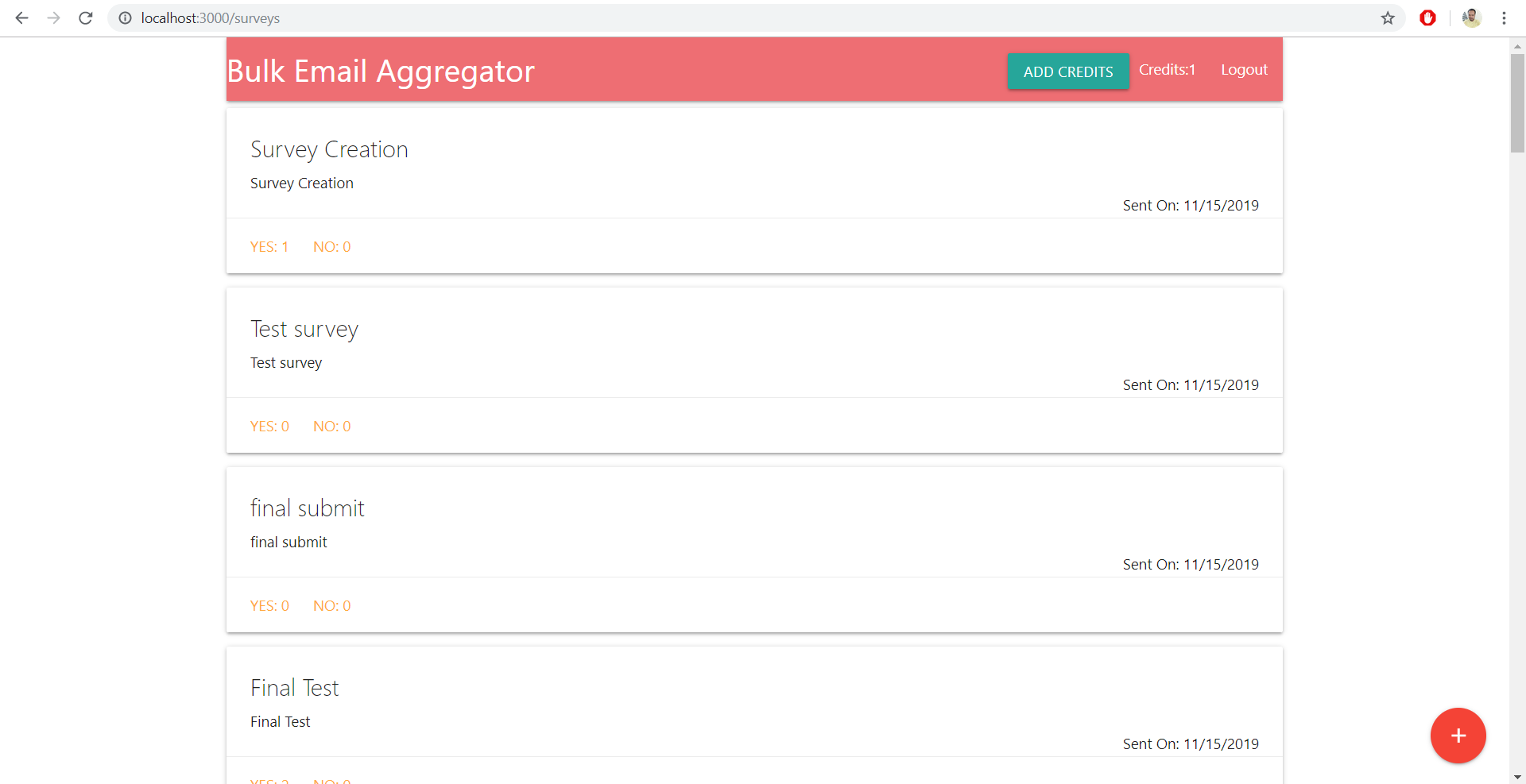
**LOGIN SCREEN**

****

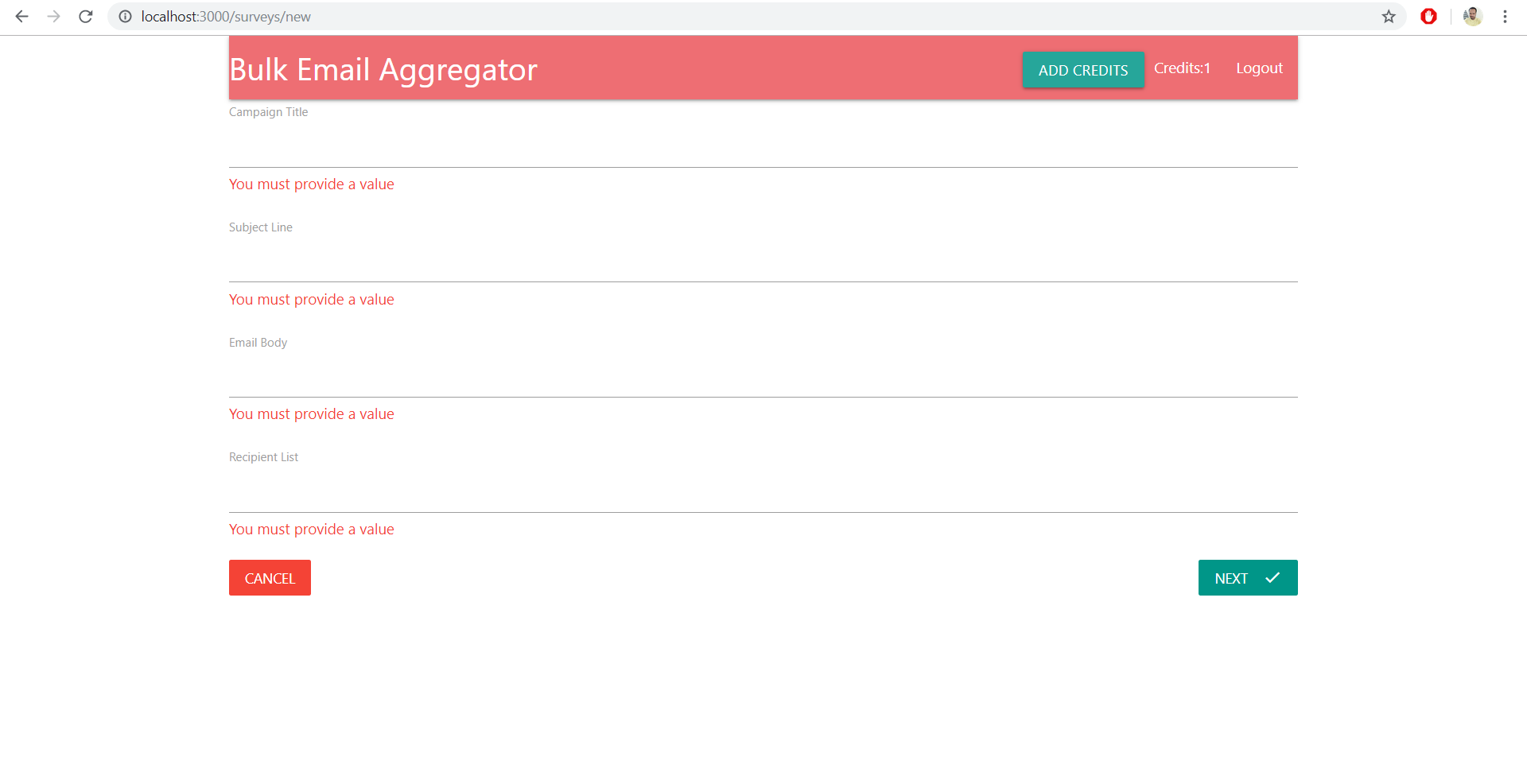
**HOMEPAGE**



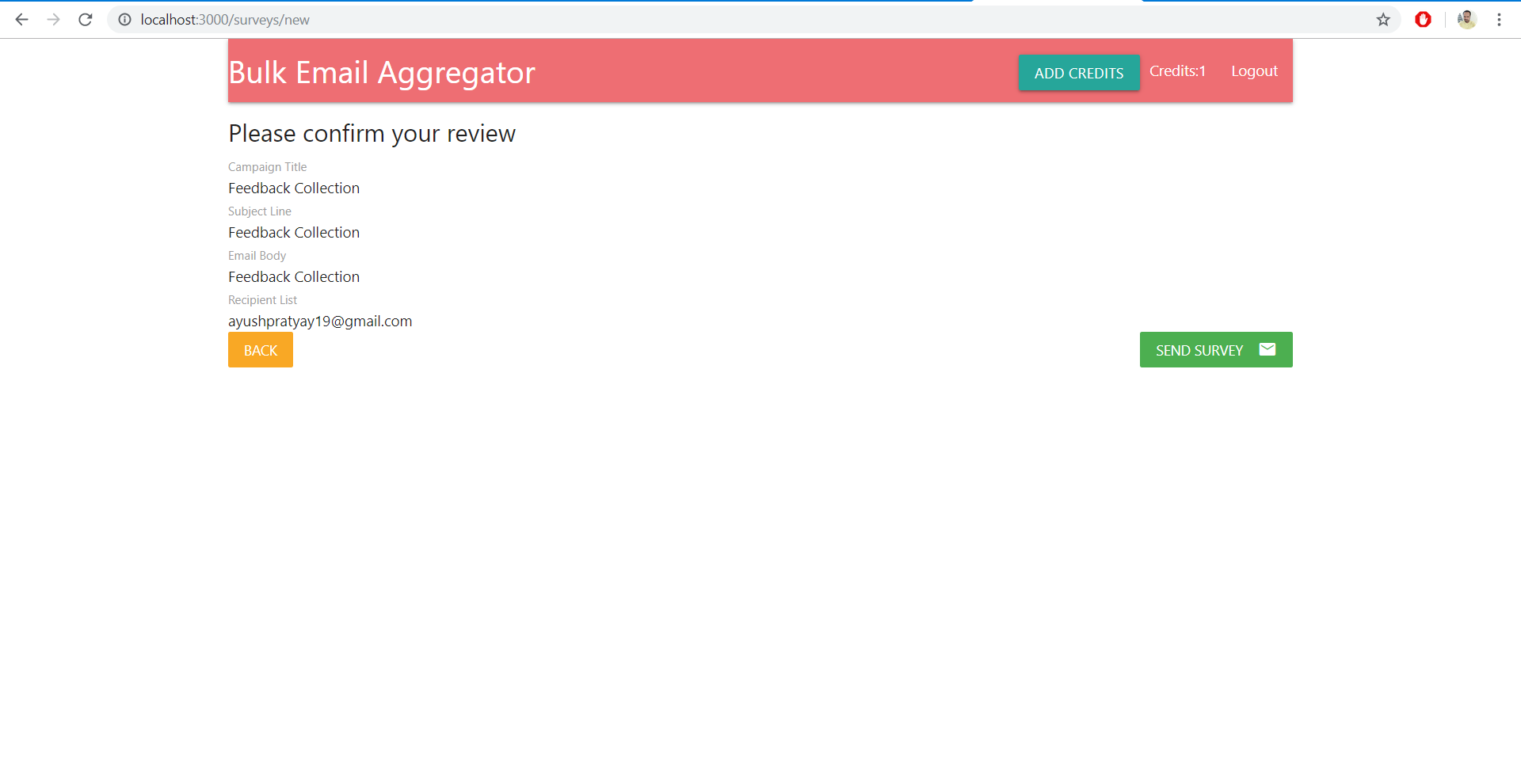
**DASHBOARD**

****

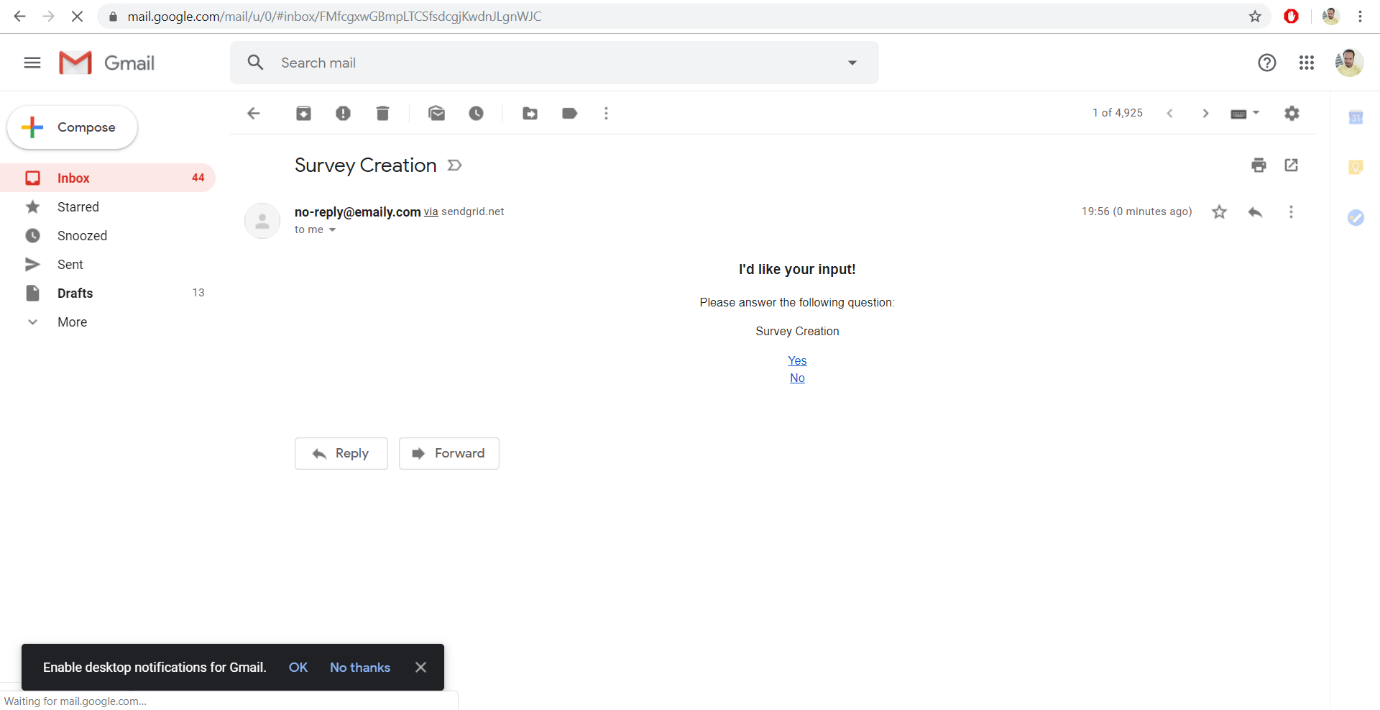
**FEEDBACK FORM**

****

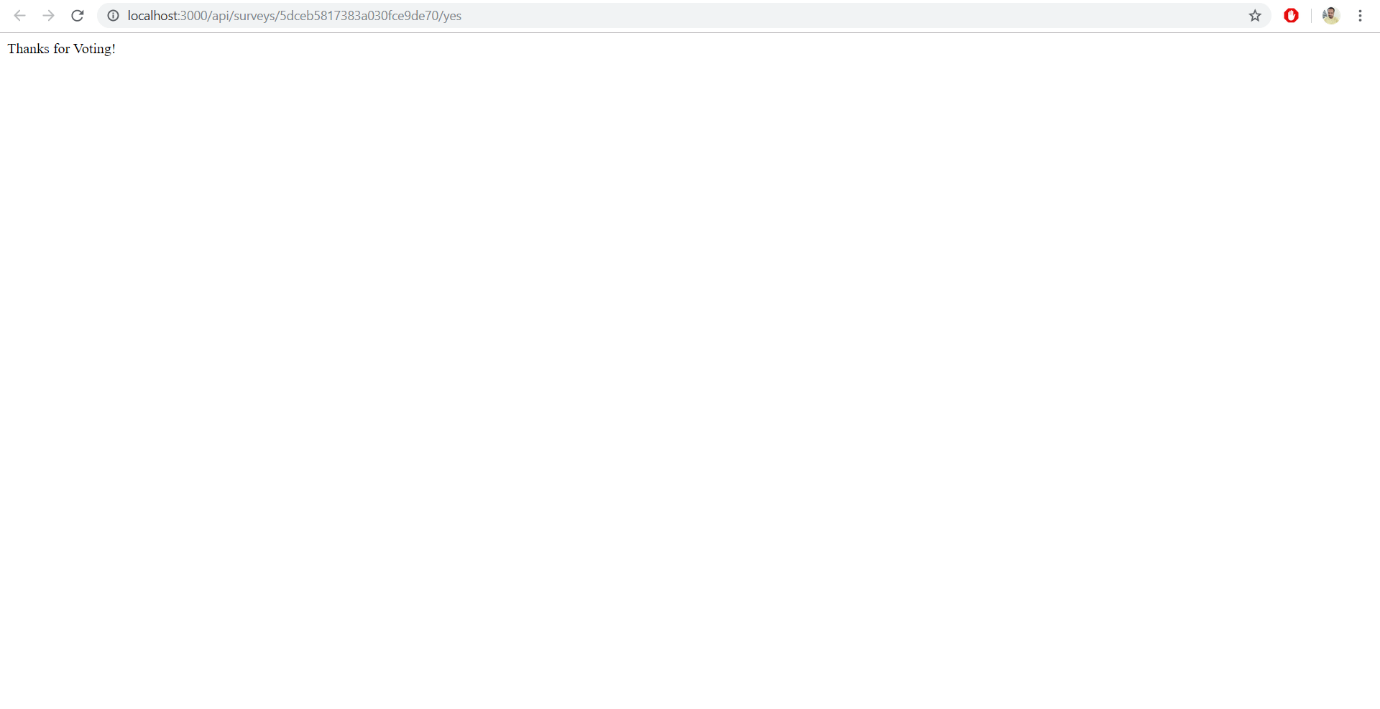
**REVIEW PAGE**



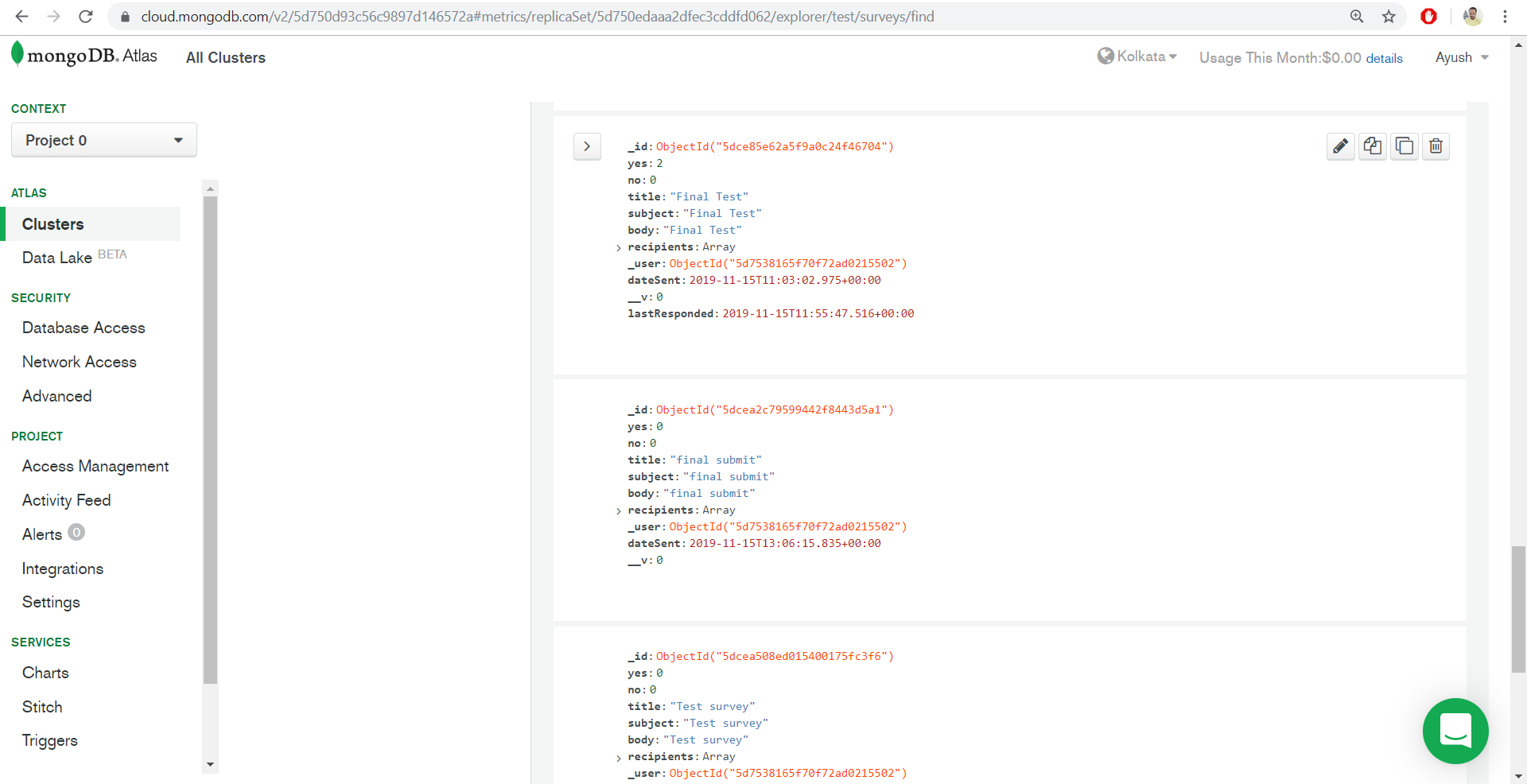
**USER RECEIVING FEEDBACK**

****

**USER RECEIVES CONFIRMATION**

****

**MONGODB STRUCTURE**



**TESTING**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | PES\_001 | **Test Case Description** | | Test the Login Functionality in Admin Panel | | | | | | |
| **Created By** | | Vijaykumar R Pai | **Reviewed By** | | Deepthi S Narayan | | | **Version** | | 1 | |
|  |  |  |  |  |  |  |  | |  |  |  | |
| **QA Tester’s Log** | |  | | | |  |  | |  |  |  | |
|  |  |  |  |  |  |  |  | |  |  |  | |
| **Tester's Name** | | Vijaykumar R Pai | **Date Tested** | | 30-Mar-2019 | | | **Test Case (Pass/Fail/Not Executed)** | | 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 |  | | | | | |
|  |  |  |  |  |  |  | |  | | |  |
| **Test Scenario** | Verify on entering valid email and password, the admin can login | | | | | |  | |  |  | |
|  |  |  |  |  |  |  | |  | | |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | | | | **Pass / Fail / Not executed / Suspended** |
|
| 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 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | PES\_002 | **Test Case Description** | | Test the Registration form Functionality in Admin Panel | | | | | | |
| **Created By** | | Vijaykumar R Pai | **Reviewed By** | | Deepthi S Narayan | | | **Version** | | 1 | |
|  |  |  |  |  |  |  |  | |  |  |  | |
| **QA Tester’s Log** | |  | | | |  |  | |  |  |  | |
|  |  |  |  |  |  |  |  | |  |  |  | |
| **Tester's Name** | | Vijaykumar R Pai | **Date Tested** | | 30-Mar-2019 | | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | | |
| 1 | Access to Chrome Browser | | |  | 1 | Name = Vijaykumar R Pai | | | | | |
| 2 |  | | |  | 2 | E-mail = [vijaykumarrpai@gmail.com](mailto:vijaykumarrpai@gmail.com) | | | | | |
| 3 |  | | |  | 3 | Mob-no = 7582455420 | | | | | |
| 4 |  | | |  | 4 | Your Address = Blore | | | | | |
|  |  | | |  | 5 | Password = 12345678 | | | | | |
|  |  | | |  | 6 | Confirm Password = 12345678 | | | | | |
|  |  |  |  |  |  |  | |  | | |  |
| **Test Scenario** | 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 executed / Suspended** |
|
| 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 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | PES\_003 | **Test Case Description** | | Test adding new package Functionality in Admin Panel | | | | | | |
| **Created By** | | Vijaykumar R Pai | **Reviewed By** | | Deepthi S Narayan | | | **Version** | | 1 | |
|  |  |  |  |  |  |  |  | |  |  |  | |
| **QA Tester’s Log** | |  | | | |  |  | |  |  |  | |
|  |  |  |  |  |  |  |  | |  |  |  | |
| **Tester's Name** | | Vijaykumar R Pai | **Date Tested** | | 30-Mar-2019 | | | **Test Case (Pass/Fail/Not Executed)** | | 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 | | | | | |
|  |  |  |  |  |  |  | |  | | |  |
| **Test Scenario** | 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 executed / Suspended** |
|
| 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 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | PES\_004 | **Test Case Description** | | Test adding Package list Functionality in Admin Panel | | | | | | |
| **Created By** | | Vijaykumar R Pai | **Reviewed By** | | Deepthi S Narayan | | | **Version** | | 1 | |
|  |  |  |  |  |  |  |  | |  |  |  | |
| **QA Tester’s Log** | |  | | | |  |  | |  |  |  | |
|  |  |  |  |  |  |  |  | |  |  |  | |
| **Tester's Name** | | Vijaykumar R Pai | **Date Tested** | | 30-Mar-2019 | | | **Test Case (Pass/Fail/Not Executed)** | | 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 | | | | | |
|  |  |  |  |  |  |  | |  | | |  |
| **Test Scenario** | Verify on clicking the delete button in package list, the respective package is deleted. | | | | | |  | |  |  | |
|  |  |  |  |  |  |  | |  | | |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | | | | **Pass / Fail / Not executed / Suspended** |
|
| 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 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | PES\_005 | **Test Case Description** | | Test Agent list Functionality in Admin Panel | | | | | | |
| **Created By** | | Vijaykumar R Pai | **Reviewed By** | | Deepthi S Narayan | | | **Version** | | 1 | |
|  |  |  |  |  |  |  |  | |  |  |  | |
| **QA Tester’s Log** | |  | | | |  |  | |  |  |  | |
|  |  |  |  |  |  |  |  | |  |  |  | |
| **Tester's Name** | | Vijaykumar R Pai | **Date Tested** | | 30-Mar-2019 | | | **Test Case (Pass/Fail/Not Executed)** | | 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 | | | | | |
|  |  |  |  |  |  |  | |  | | |  |
| **Test Scenario** | Verify on clicking the delete button in package list, the respective package is deleted. | | | | | |  | |  |  | |
|  |  |  |  |  |  |  | |  | | |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | | | | **Pass / Fail / Not executed / Suspended** |
|
| 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 / Suspended** | | |
|
| 1 | Navigate to <http://localhost:3000/> | | 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 / Suspended** | | |
|
| 1 | Navigate to <http://localhost:3000/> | | 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 / Suspended** | |
|
| 1 | Navigate to <http://localhost:3000/> | | 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 CRM’s who have to send individual emails to each customer who purchase products from different electronic companies, which has been solved with our application as it provides a paid service to end users which will enable them to send bulk emails for collecting feedback of their own electronic products once they complete transaction through Stripe payment gateway. Our service will be provided on usage of assigned credit points to user. So once the CRM who will be using the application does payment, 5 credit points will be incremented to the user’s account and he can send emails with this credit. Since our application is built with React.js, it is lightweight, responsive, loads faster once it will be deployed. In addition, the whole application has been deployed on Heroku platform. So in the future, if the user requests for any changes, it can be easily done through git version control. Hence, the objective has been fulfilled and application provides bulk email delivery.

**BIBLIOGRAPHY**

1. https://www.coursera.com/
2. <https://reactjs.org/tutorial/tutorial.html>
3. <https://www.w3schools.com/react/>
4. <https://www.npmjs.com/>
5. <https://developers.google.com/identity/protocols/OAuth2>
6. <https://stripe.com/docs/api>
7. <https://devcenter.heroku.com/categories/nodejs-support>
8. <https://sendgrid.com/docs/for-developers/>
9. <https://www.edx.org/>
10. <https://docs.mongodb.com/cloud/>