1. **INTRODUCTION**

**OVERVIEW**

This report discusses the result of the work done in development of “Caption Generator” on Machine Learning Platform. The project aims at the development of an application that identifies the action portrayed in the given image.

**BACKGROUND AND MOTIVATION**

**Existing Systems**

* **Captionbot.ai**
  + It is a product of Microsoft.
  + It is an ML application that can understand the content of any image.
  + When a person upload a photo, it is sent to Microsoft for image analysis to return a caption
  + The application will not store or publish the images anywhere.
  + It uses Computer Vision API and Emotion API
* **How-Old.net**
  + It is also a product of Microsoft.
  + It estimates the age of the person in the given image.
  + The age is generated as a caption
* **TwinsOrNot.net**
  + It identifies whether the image has twins or not

**OBJECTIVE**

The final goal of the project is twofold.

1. A
2. ddwaddadawda

**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 python packages and Machine Learning models.
3. Specifying the server ports between the modules of flask.
4. Analysis: Extracting the required data for analysis and then doing the analysis.

**ANALYSIS**

Based on analysis and literature survey regarding the present difficulties faced by the existing systems like Captionbot.ai, How-Old.net and TwinsOrNot.net as Captionbot.ai do not allow users to upload images as per their desires. How-Old.net does not give accurate predictions for all uploaded images and TwinsOrNot.net is a deprecated system.

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.

**REQUIREMENTS ANALYSIS**:

**SOFTWARE REQUIREMENTS**:

Operating System: Windows 10 / Ubuntu

Front end: HTML, CSS, Bootstrap

Back end: Flask 1.x

Language: Python 3.7

Storage (Dataset): Google Drive

IDE: Jupyter Notebook

Cloud Deployment: Heroku Deployment

Other Technologies used: Git and GitHub

**HARDWARE REQUIREMENTS:**

RAM: 8GB and above

Hard disk: 120GB and above

Processor: Intel i3 and above

**FUNCTIONALITIES**:

* **User Perspective**
  + User launches the application.
  + User uploads the images for which the captions are to be generated.
  + User clicks on the submit button.
  + Within a matter of seconds, the ML model recognizes and process the content of an image.
  + The result will be displayed on the page.
* **Model Development and Deployment**
  + Data Gathering
    - Images for training the model
    - Corpus for Image captions
  + Pre-processing
    - Corpus – Removal of stop words, punctuation marks, digits
    - Generating the bag of words
    - Mapping of images with the Corpus
  + Feature extraction from images
  + Object identification
  + Model Generation using CNN
  + Building LSTM (Long Short Term Memory) model.
  + Validating the Model
  + Deploy the ML model in the web application

**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**

**DFD**

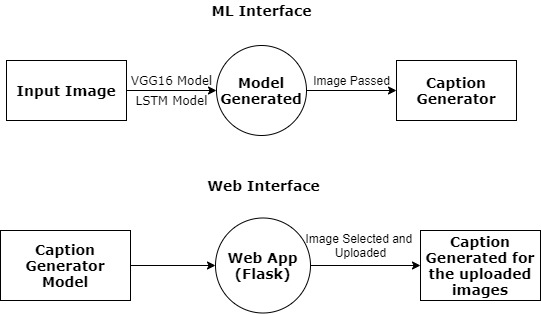


Figure 3.1.1 DFD Level 0

Our project consists of only one fold:

1. Visually impaired users: Users who would like to generate captions for the images that they had captured.

**PROCESS FLOW DIAGRAM**

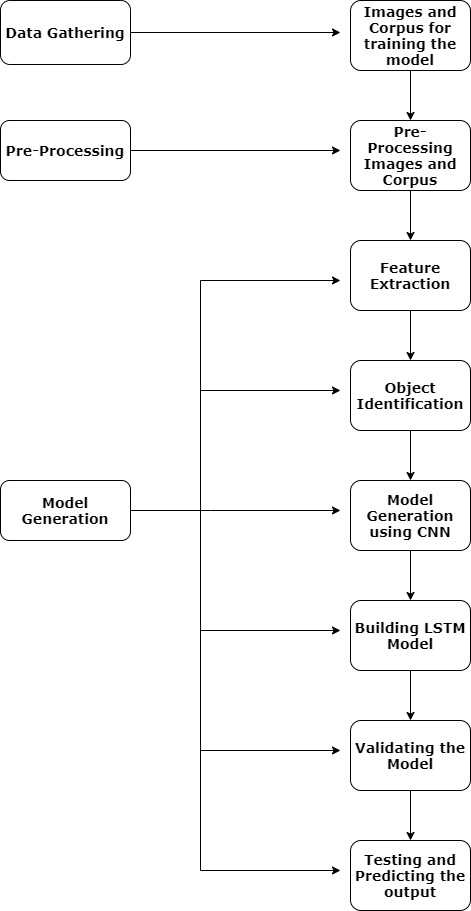


Figure 3.1.2 Use Case Diagram

**SCREEN SHOTS**

**LOGIN SCREEN**

****

Figure 4.1.1 Login Screen

**HOMEPAGE**

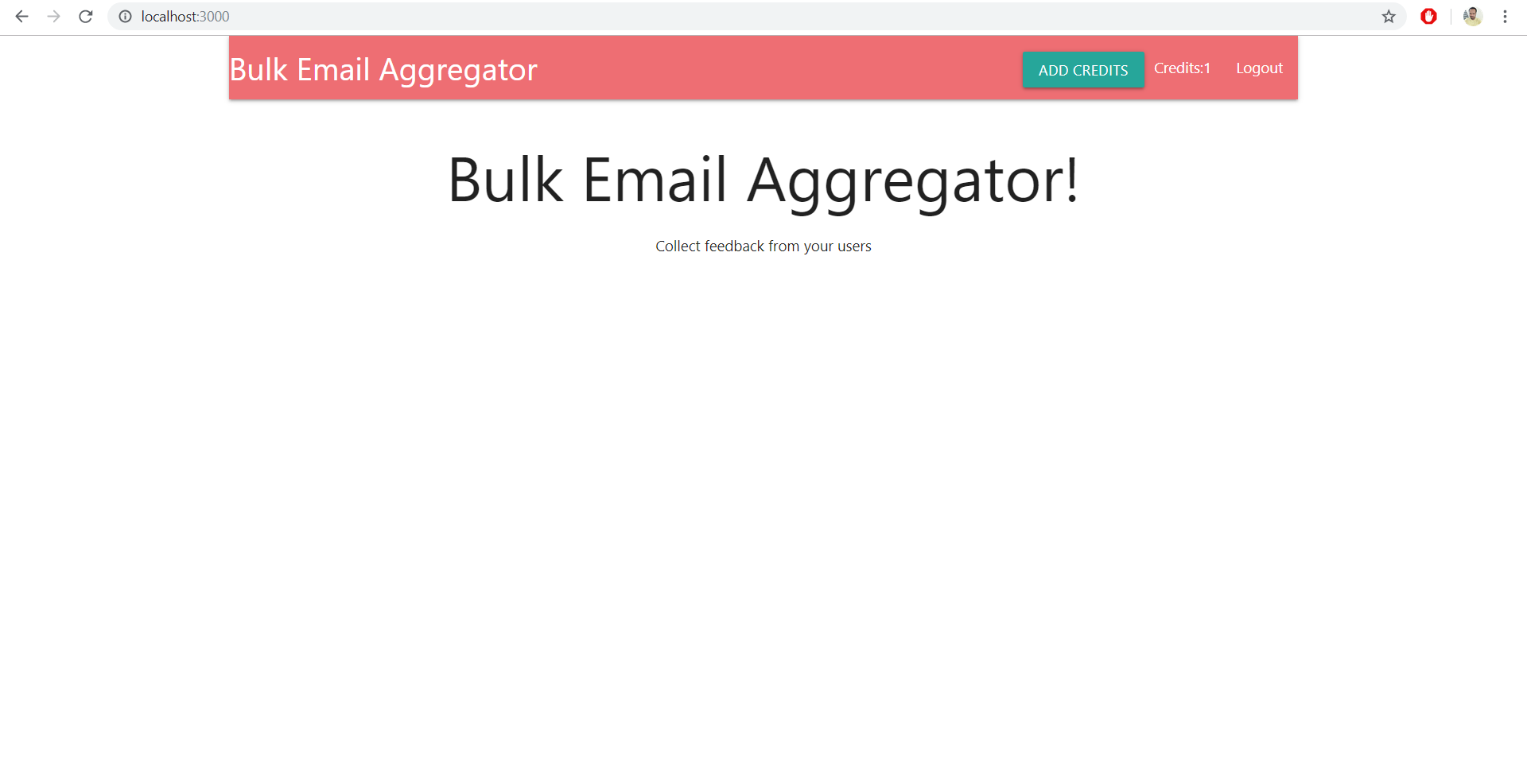


Figure 4.1.2 Home Page Screen

**DASHBOARD**

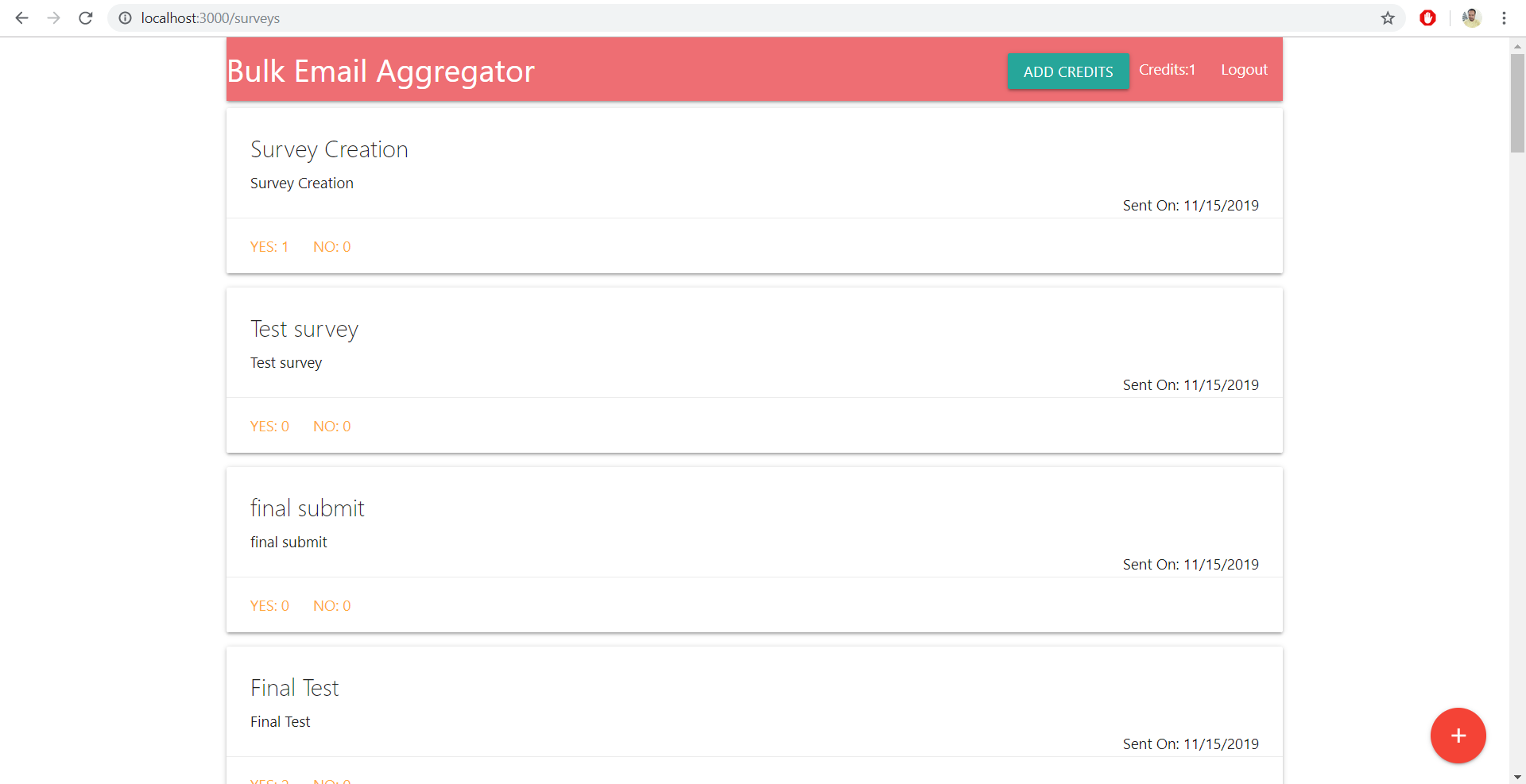
****

Figure 4.1.3 Dashboard

**FEEDBACK FORM**

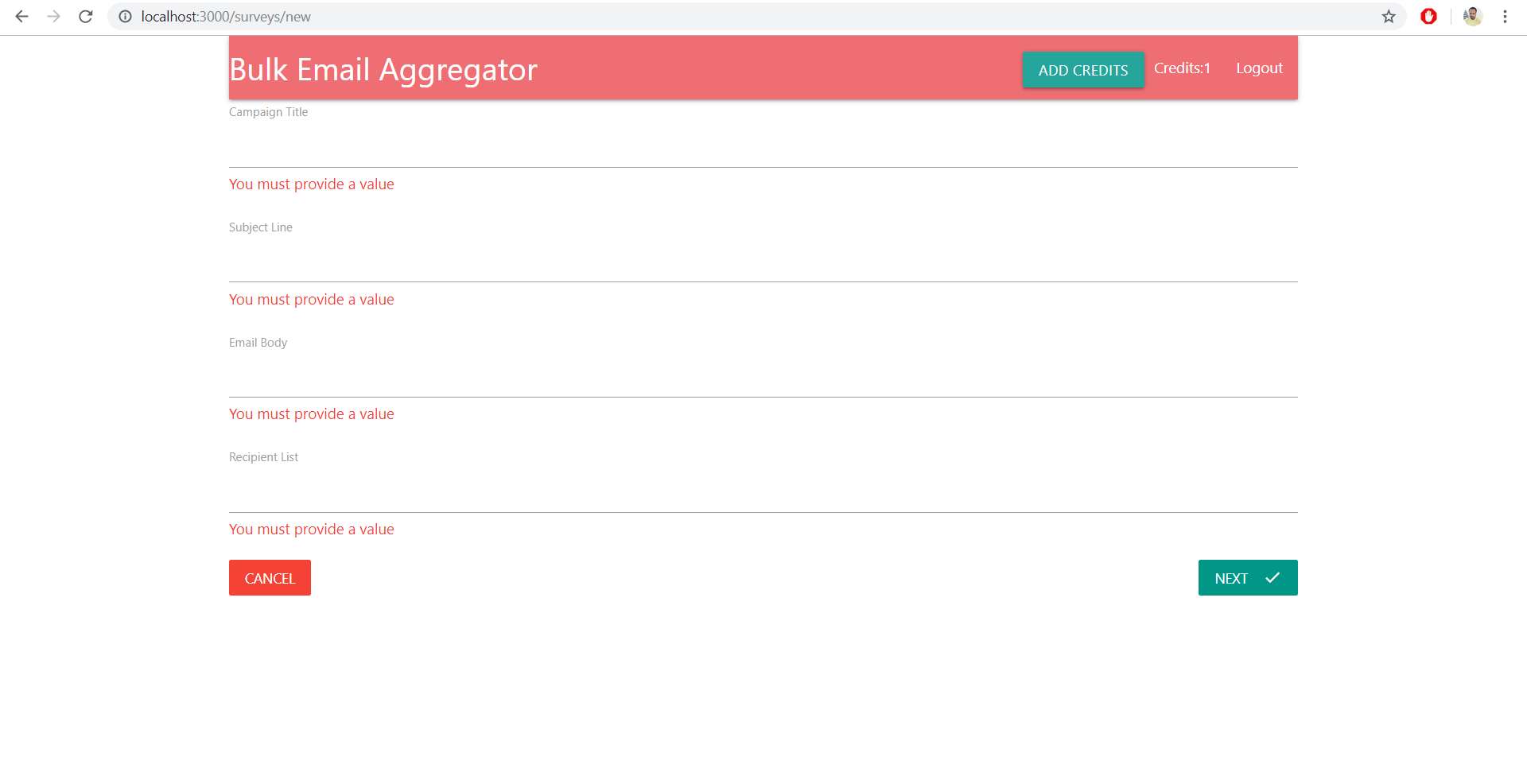
****

Figure 4.1.4 Feedback Form

**REVIEW PAGE**

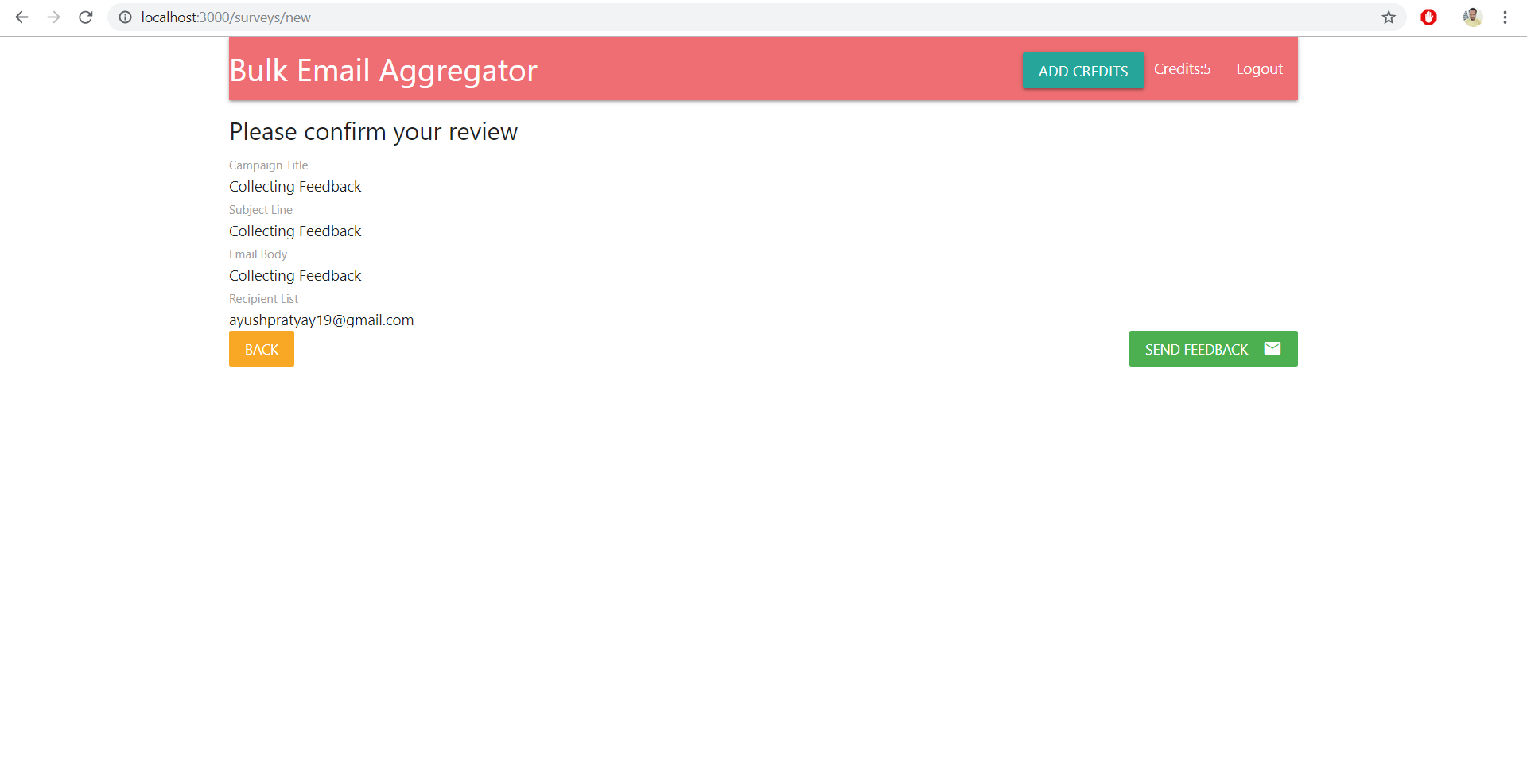


Figure 4.1.5 Review Page

**USER RECEIVING FEEDBACK**

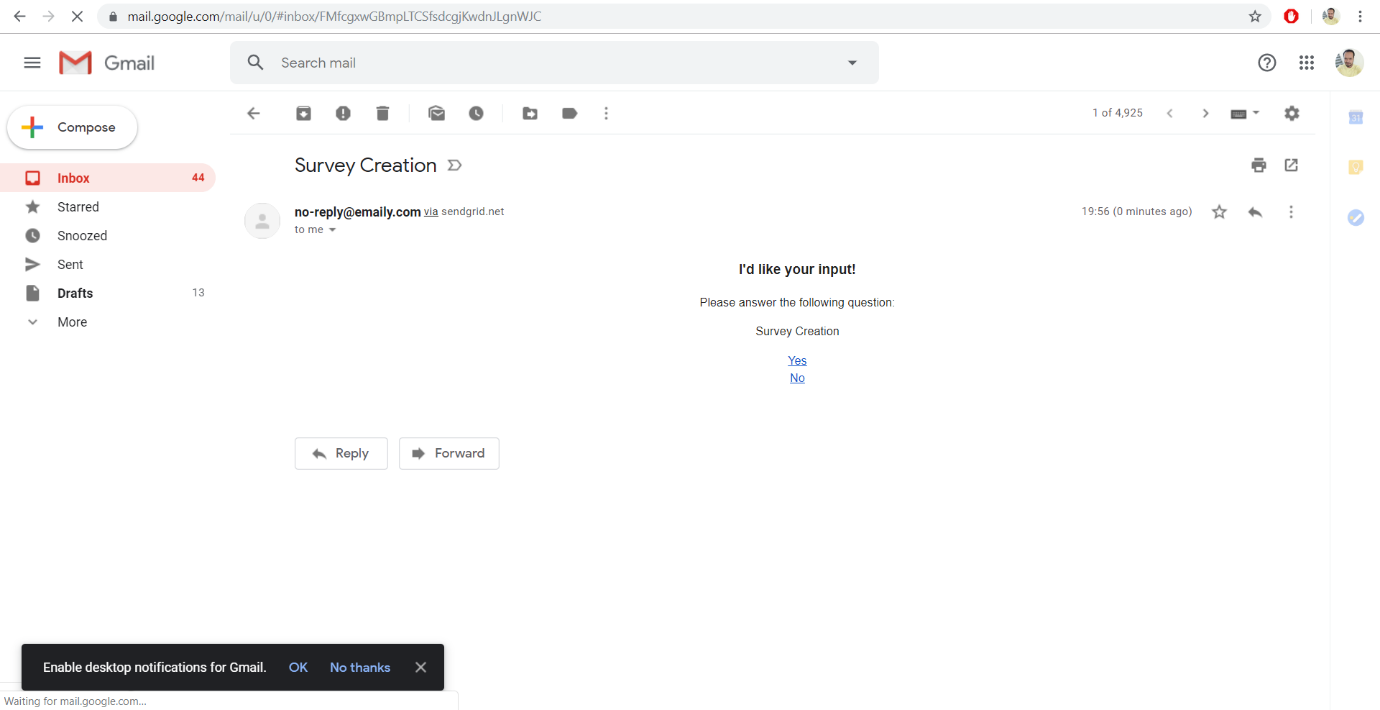
****

Figure 4.1.6 User Receiving Feedback

**USER RECEIVES CONFIRMATION**

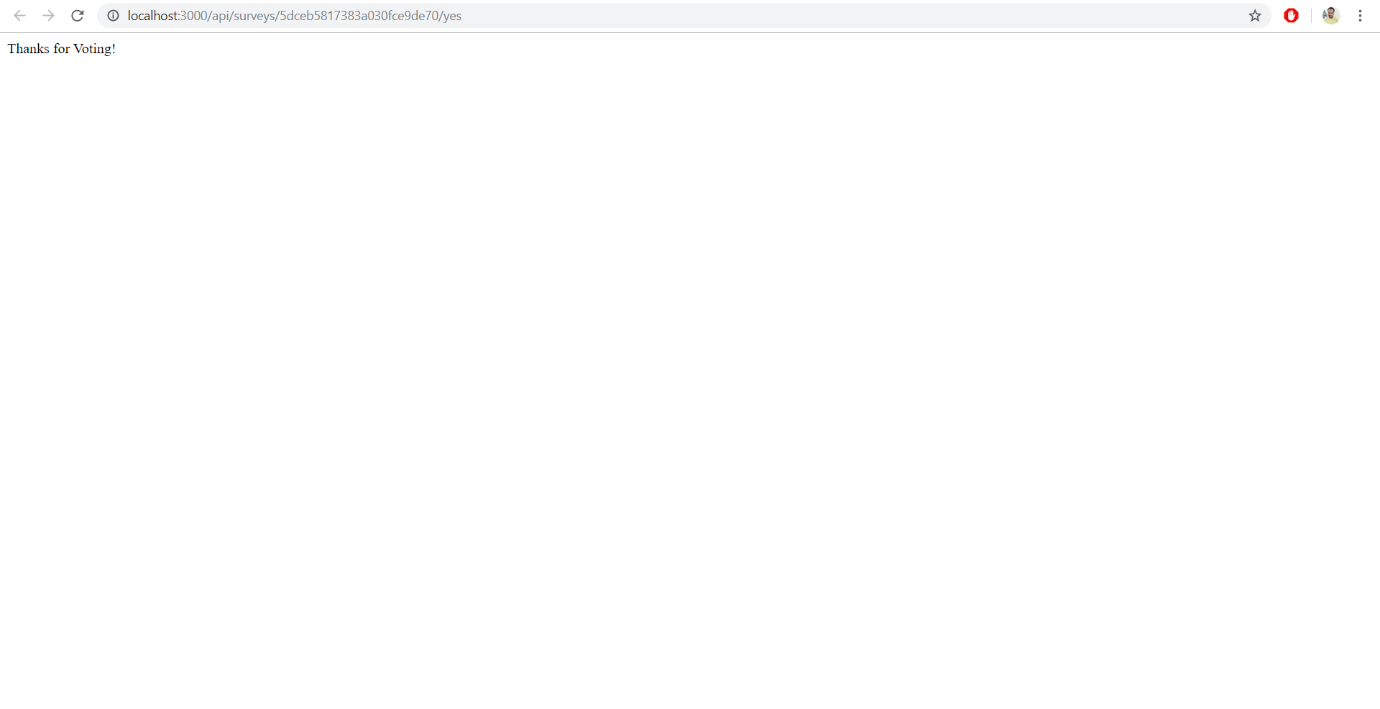
****

Figure 4.1.7 User Receives Confirmation

**STRIPE**

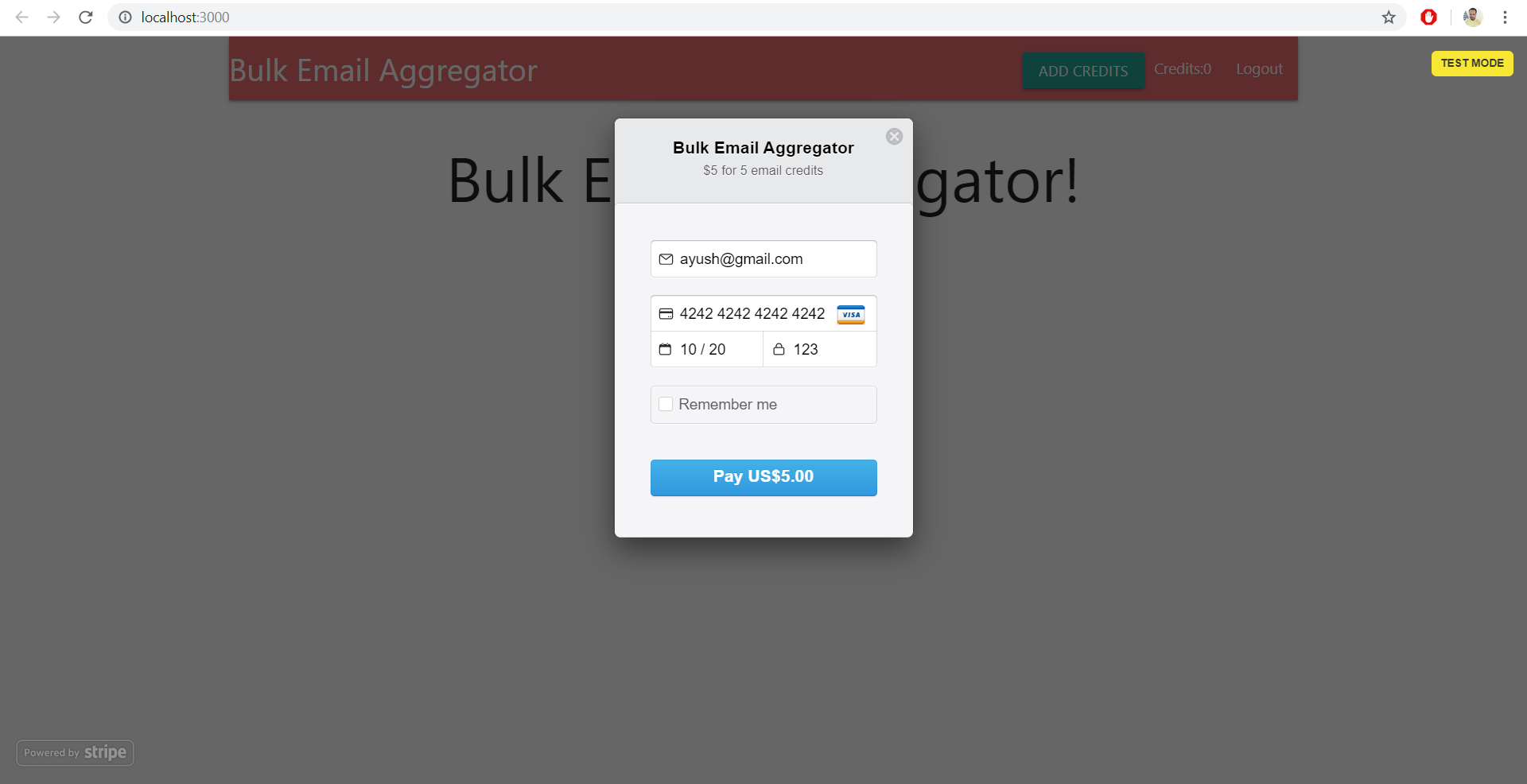
****

Figure 4.1.8 Stripe Payment Gateway

**MONGODB STRUCTURE**

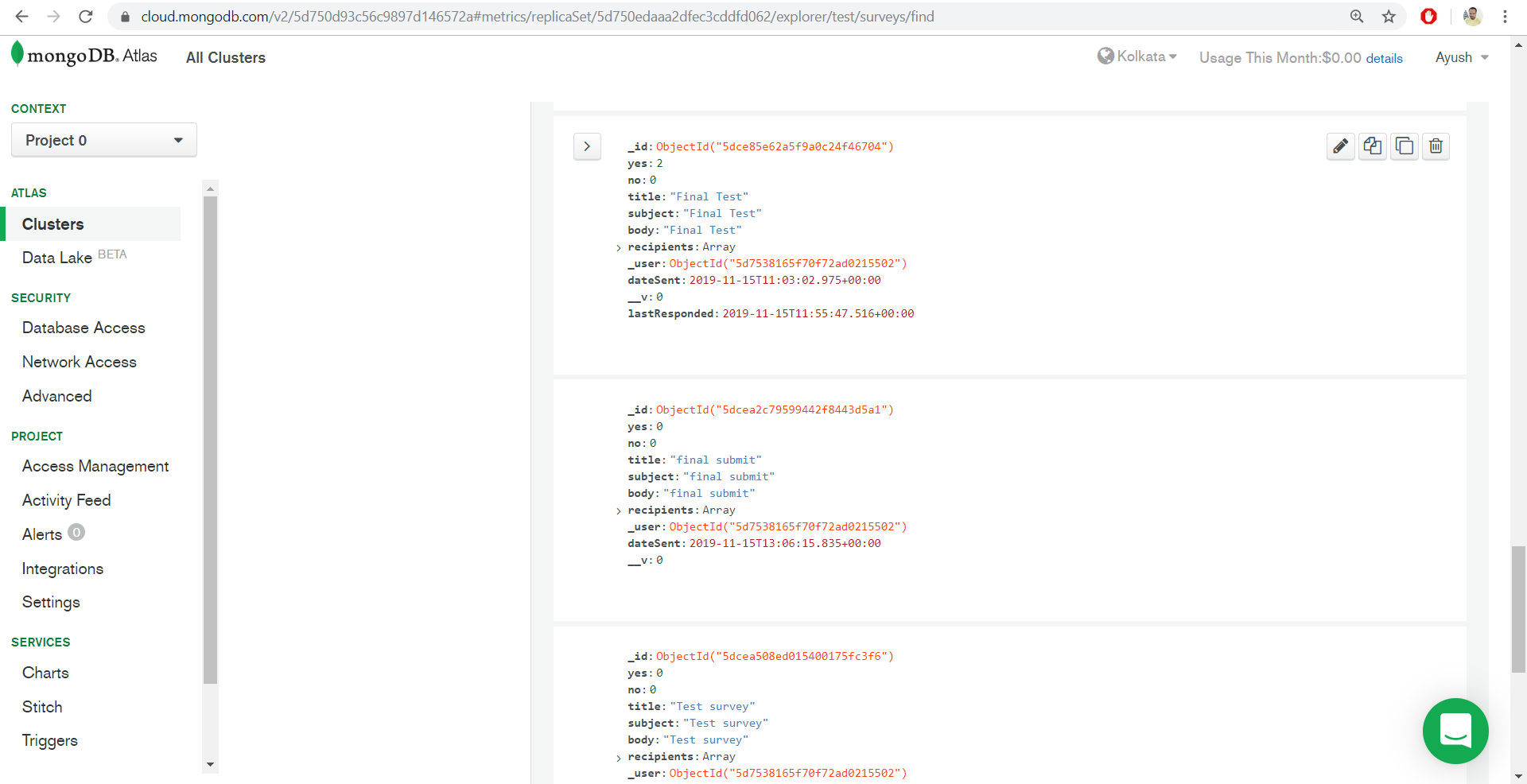


Figure 4.1.8 MongoDB Atlas

**TESTING**

Table 5.1 Test Case

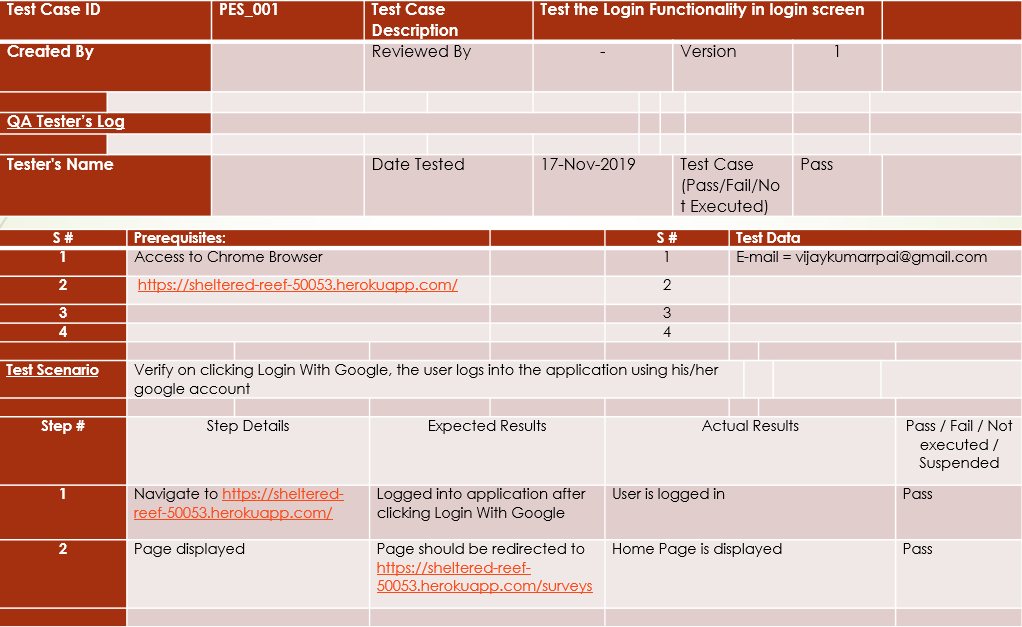
****

Table 5.2 Test Cases

****

Table 5.3 Test Case

****

Table 5.4 Test Case

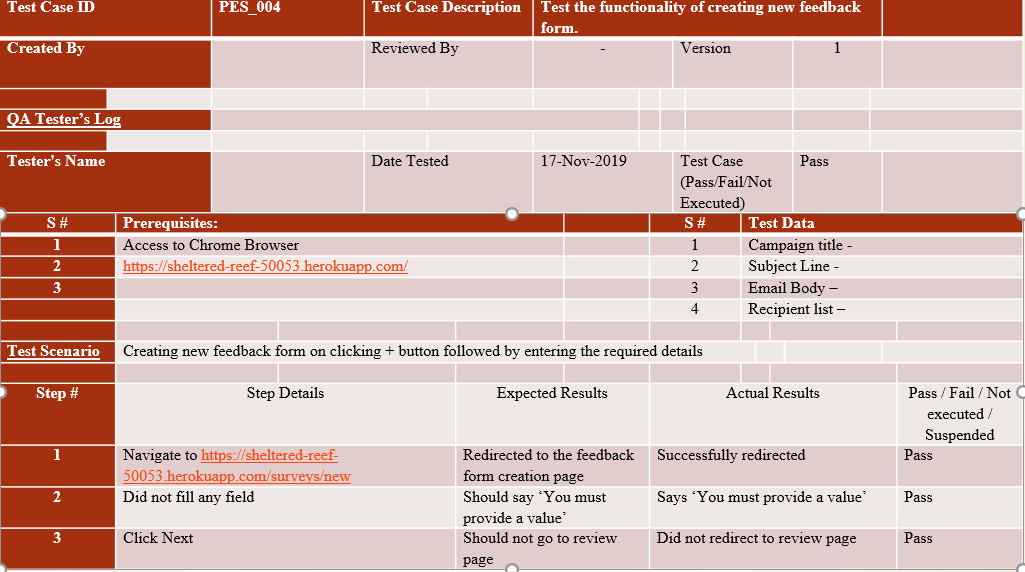
****

Table 5.5 Test Case

****

Table 5.6 Test Case



Table 5.7 Test Case

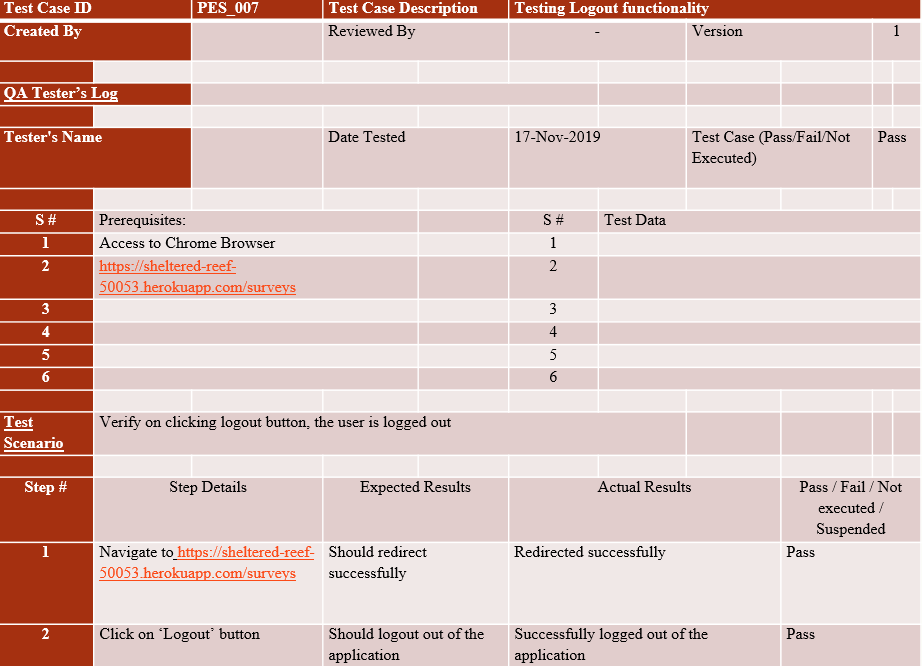
****

Table 5.7 Test case

**CONCLUSION**

1. The objective of the project was to solve the difficulties faced by the CRM’s who have to send individual emails to each customer.
2. This has been solved with our application as it provides a paid service to CRM’s which will enable them to send bulk emails for collecting feedback of their own electronic products.
3. 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.

**FUTURE ENHANCEMENT**

1. Lot of features and functionalities can be integrated in our project. Firstly, we can group certain customers into one batch so that we can send bulk emails in one shot.
2. Secondly, we can build a customized Mobile app which will make user more convenient to use.

**BIBLIOGRAPHY**

1. Book on **Deep Learning for Natural Language Processing** by **Jason Brownlee**

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