

README

PROJECT NAME: FEEDBACK MACHINE

Synopsis:

We have developed an application, 'Feedback Machine' that acts as a medium for taking feedbacks from the end-users. This application is especially for the startup owners or managers or any organization requiring feedbacks from their customers for improving their services and increasing the overall value of their product.

Technologies:

The application is implemented using the MVC framework.

For the Models, that is for the database operations mlab is used which is a cloud database service hosting MongoDB database. (<https://mlab.com>)

The front-end that is the View part is implemented using ReactJS and Redux.

CSS was used for styling. (Used materialize-css package for styling our application.)

The user can sign in using google, which is done using OAuth 2.0.

The Controller part was done using NodeJS and Express. For authentication, PassportJS was used.

A third party's API – SendGrid, was used for sending and receiving e-mails. Web pack helped in bundling all the dependencies.

The payment is managed using Stripe, another third-party application API.

As of now, the application is in the test mode, so for any addition of the credits, any random email ID, expiry date and CVV is acceptable.

The card number to be put is 4242 4242 4242 4242 for the test mode.

Functionalities:

We start the server. The homepage is displayed first.

We can sign in using google. [Need to enter our google credentials].

Once user signs in, into our application, data of that user will be getting stored in our MongoDB database.

Account Details for accessing mlab (<https://mlab.com>):

[username: dark_knights

Password: cs554@Stevens

Once you login to mlab account, click over ds133746/cs554thedarkknights-dev (sandbox environment)

You will be able to see the collections]

There are two collection that will be created in our mlab. (i.e. Mongo Database)

1) users

2) surveys

After signing in to our application, we will be able to see the dashboard where all the campaigns or surveys if any created are displayed with their respective user feedbacks and the last modified dates.

To create a new survey, we click the add button in the right-most bottom corner.

The form is filled using the required inputs. There can be any number of recipients. The recipients are separated using a comma.

After the survey is created, it can be reviewed and after reviewing, it can be sent to the recipients mentioned in the form by pressing the send survey button.

As soon as the survey is sent, respective details will get store in our surveys collection and as per the recipients input as yes or no it will be updated.

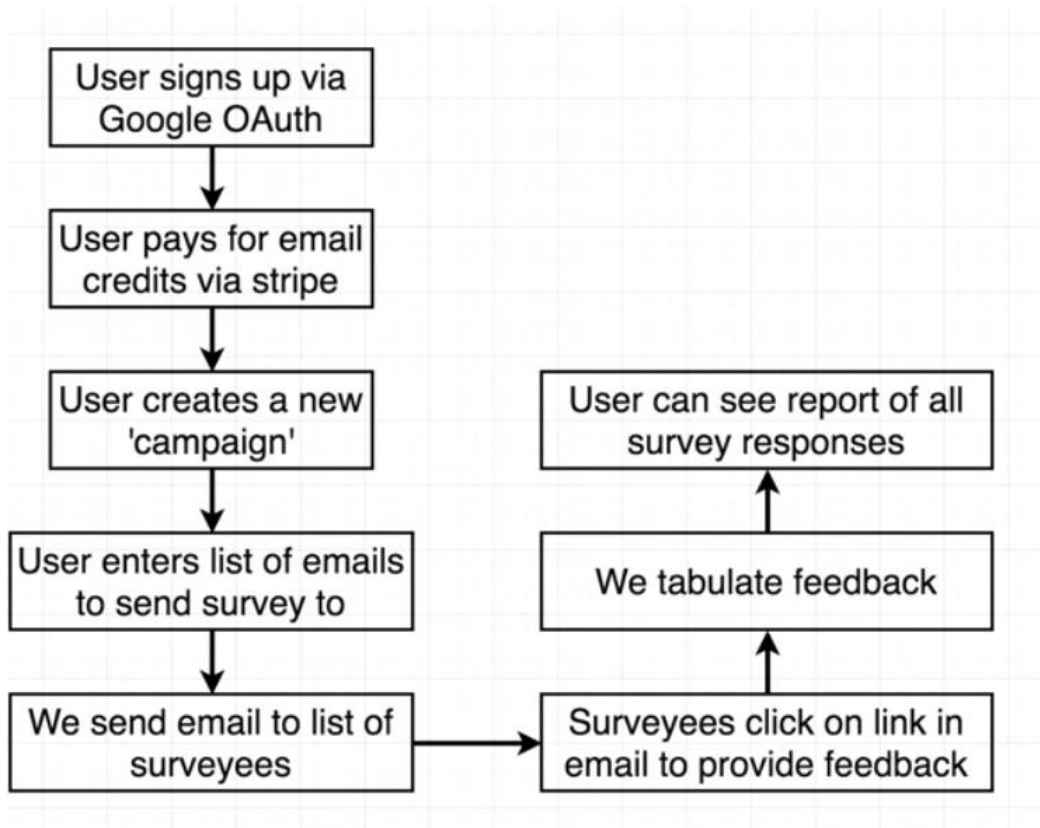
Simultaneously, the recipients will be receiving the mail.

Depending upon the options given yes or no within the received mail, they will be giving the feedback and the data is then updated in the database.

Depending upon the number of yes or no option clicked by the recipients, the survey creator will be able to see the number of feedback given as yes or no for that particular survey within his / her dashboard.

All the surveys along with their feedbacks and their counts can be viewed in the dashboard for the admin to take the required steps for the further improvisations of his product.

Flow of the project:



Commands to be followed for running the application.

1. To install all the modules plus dependencies, implement the following command
2. "npm install" in the root folder (Path: /npm install)
3. Then go to the client folder using "cd client" and install the packages used for client side using the same command "npm install"
(From '/',
type 'cd client'
Then you will be in the directory '/client'
From the client directory('/client'), type 'npm install')
4. Then we need to build the client side. Giving the client folder path implement the command "npm run build"
(From the client directory('/client'), type 'npm run build')

5. To run the project, go to the root folder and run “npm run dev”
(Go back to the root directory ‘/’ from the client directory ‘/client’,
And from the root directory ‘/’, type ‘npm run dev’)
6. Application will get run over the path - <https://localhost:3000/>

[Note: Application is tested on following system version: -

Windows 10

Node version - v8 or later

NPM version - v5.4.2

Browser – Google Chrome Version 63.0.3239.84 **or higher**

]

Contributors:

Saloni Kalsekar

Vicky Rana

Anurag Patil