FUNTIME

Project Report

https://github.com/Damandeep27/FunTime

By Damandeep Singh and Stephen Asuncion

Contents

Overview

FunTime is a website where you and your friends can meet new people in a unique way. It involves real time interactions with your friends through chatting, using different emojis movements on globally available servers. You could customize yourself to best reflect your mood or emotions. The main goal of this website is to escape from the stressful world and relax yourself by doing something simple, funny, and talking with friends.

Software Development Life Cycle (SDLC)

We chose Agile methodology and used scrum as our agile framework. As predicted things did not go well as planned always, we needed to have some changes after each sprint. As agile offers flexibility of introducing changes to fix problems at any stage, it was easy for us to add or remove things as required.
Talking about standouts, due to our work schedules, it was not possible to meet on a fixed day at the given time. We have to discuss sometimes through discord, which allowed us to response whenever we found time

Tech stack and APIs

We are using the **MERN** stack for our application. **Chara UI** is used instead of plain vanilla html for the front-end elements as Chakra UI library is simple, accessible and provides modular components for react. For authentication, we are using **Firebase** which is a Google-backed application development software helpful for creating, fixing, and managing web apps. Moreover, we are using **Stripe** for receiving payment from customers as it is quite reliable and easy to use due to well defined documentation. We are using **Socket.io** for real time interaction and chatting. Also, we are using **Twilio** for sending SMS, **MailJet** for sending Email upon successful payment and using **Postman** for testing API. Moreover, **Github** is used for build and version control and the app is deployed on **heroku**.

Application Features

- Global server: People could join from any dispersed geographic location.
- Chat with friends: People could chat with each other.
- Real time interaction: People could interact with each other with through emojis
- Buy emoji: People can customize their emoji by purchasing emojis available from the shop.
- Successful order message notification: Upon successful payment, customers receive successful payment notification.
- Successful order email notification: Upon successful payment, the customer also receives successful order placement confirmation through email.
- Authentication with Firebase Google OAuth

API Testing

All API routes are tested with Postman. All request routes of FunTime are protected by an access token from google's OAuth. You can get a new token by importing our Postman Exported Collection. Full instruction could be found in our repository, under **Manual Testing**.

CI/CD infrastructure

Integration

We used GitHub for continuous integration. Every time something was pushed to the main branch, it triggered a GitHub action which pushed the containerized code to the docker hub. To make it easy for each team member to understand changes and contribute, following practices have been used:

- Each change or feature is implemented using separate branch
- All branches are named using username + feature name (e.g. stephenasuncionDEV/payment)
- The commit messages include the brief information about changes implemented
- The pull requests are resolved using issue number
- The updated main branch directly deploys to Heroku

Deployment

We used Heroku for continuous deployment. Once a branch is merged with the main branch on our repository, it automatically deploys to Heroku.

PCI compliance

We are using stripe API to handle payments. Stripe has been audited by an independent PCI Qualified Security Assessor (QSA) and is certified as a PCI Level 1 Service Provider. (Source: https://stripe.com/docs/security/guide)
All the payment information is being handled by stripe; no payment information is stored on the application database. We are using one of recommended payments integrations by Stripe to collect payment information, which is securely transmitted directly to Stripe without it passing through our servers. Secondly, we are using Transport Layer Security protocol (TLS) to prevent middleman from accessing and manipulating any sensitive information while user is accessing our website or making payments.

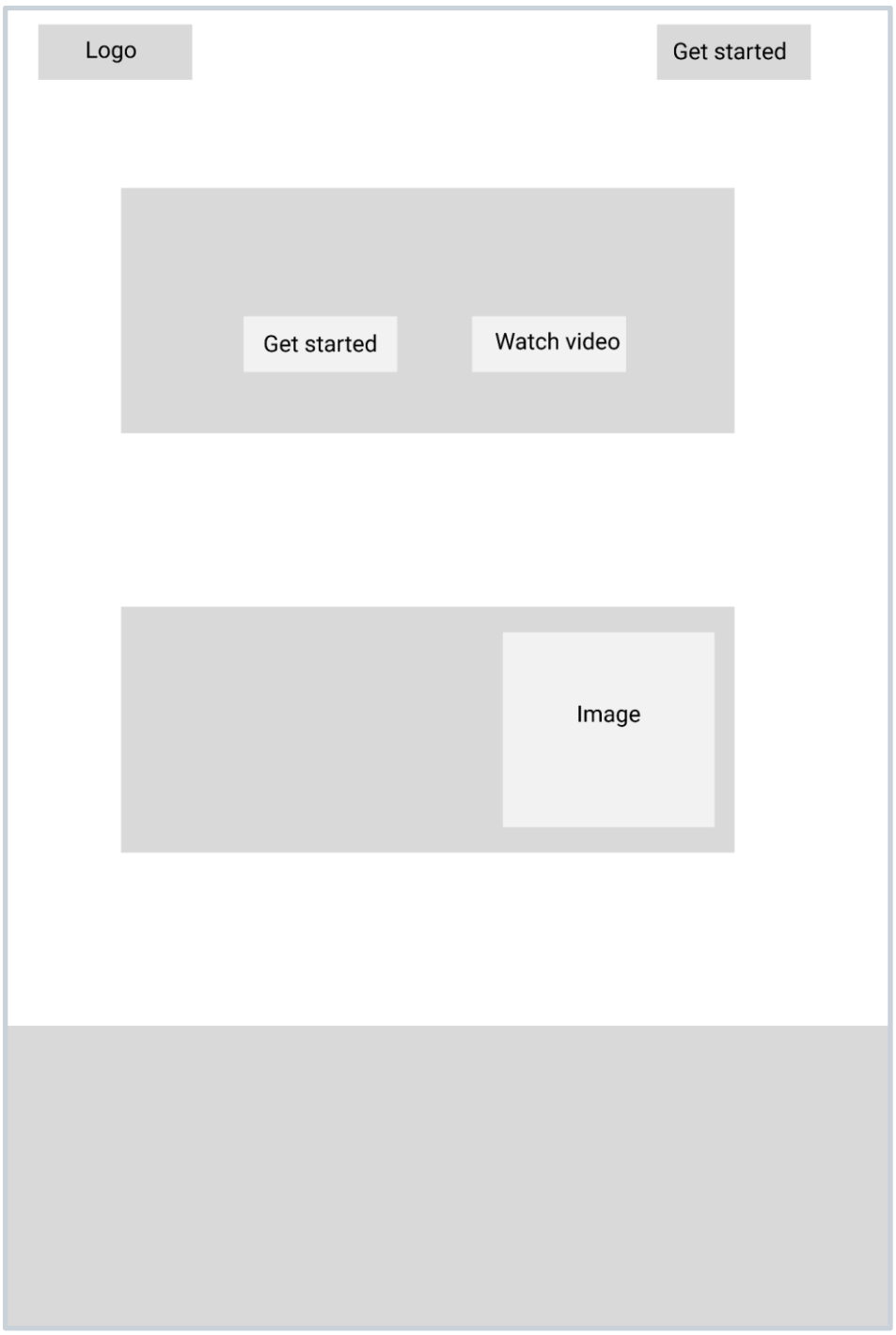
Accessibility

We are using Chakra UI that provides us with accessible elements. List of web accessibility efforts by FunTime:

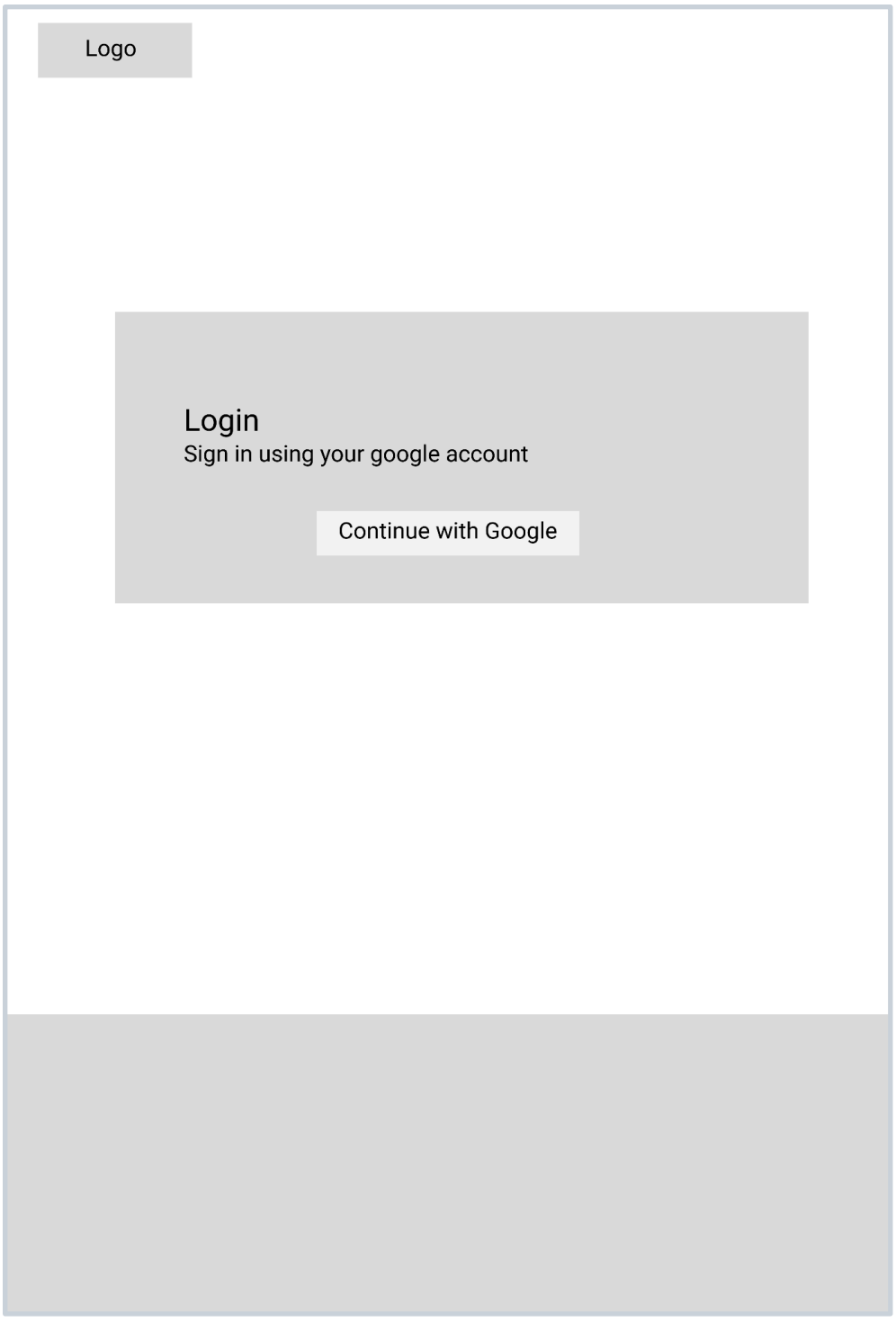
- All html elements have required attributes and valid values
- Semantics elements are used carefully to allow screen reader to interpret correctly
- The font style and size are carefully chosen
- There’s a proper contrast between background color and font color, etc.
- Buttons have an accessible name
- Image elements have [alt] attributes
- Document has a meta description
- Every page has successful HTTP status code
- Every page has a <meta name="viewport"> tag with width or initial scale

Wireframes

Home

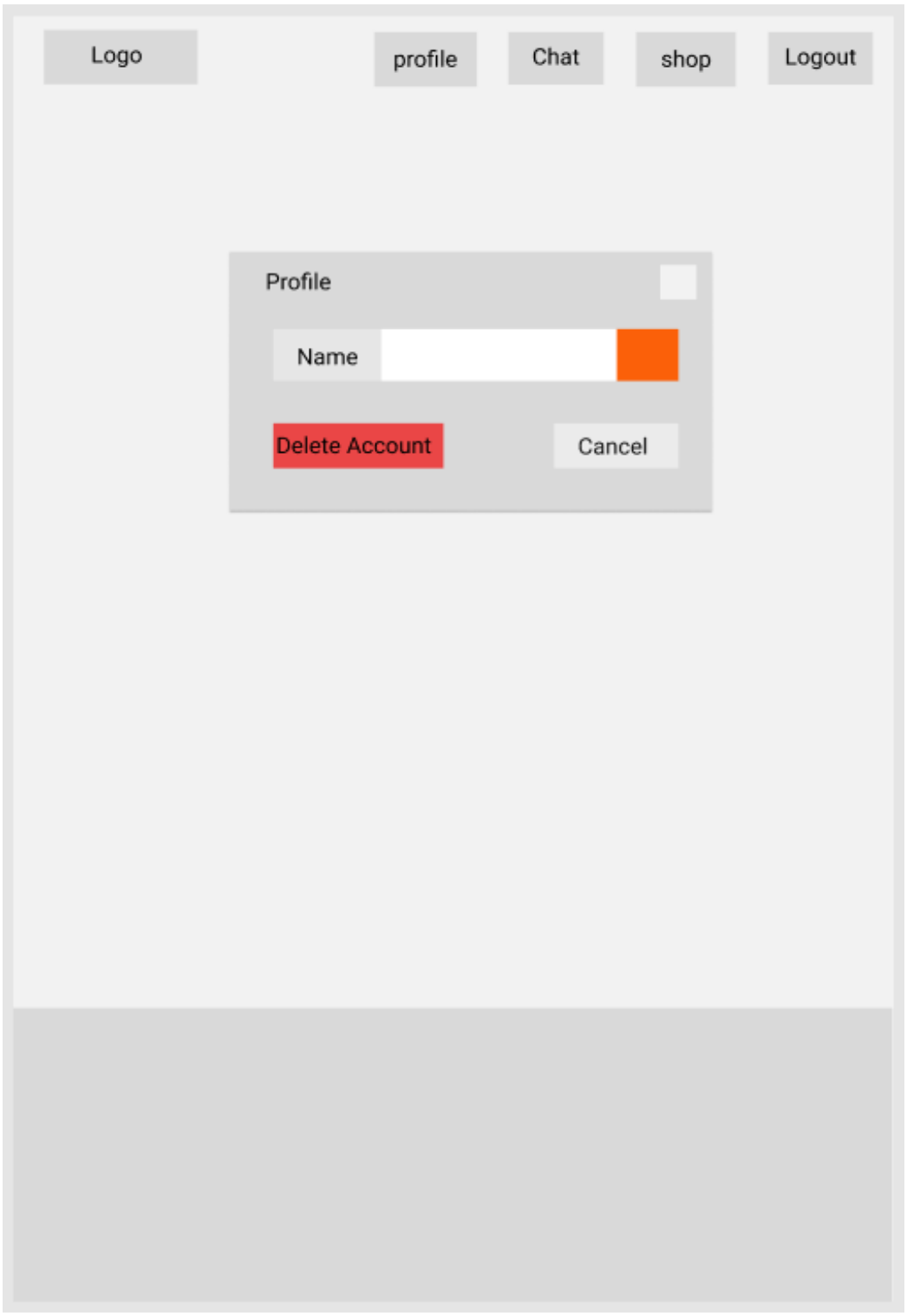


Login

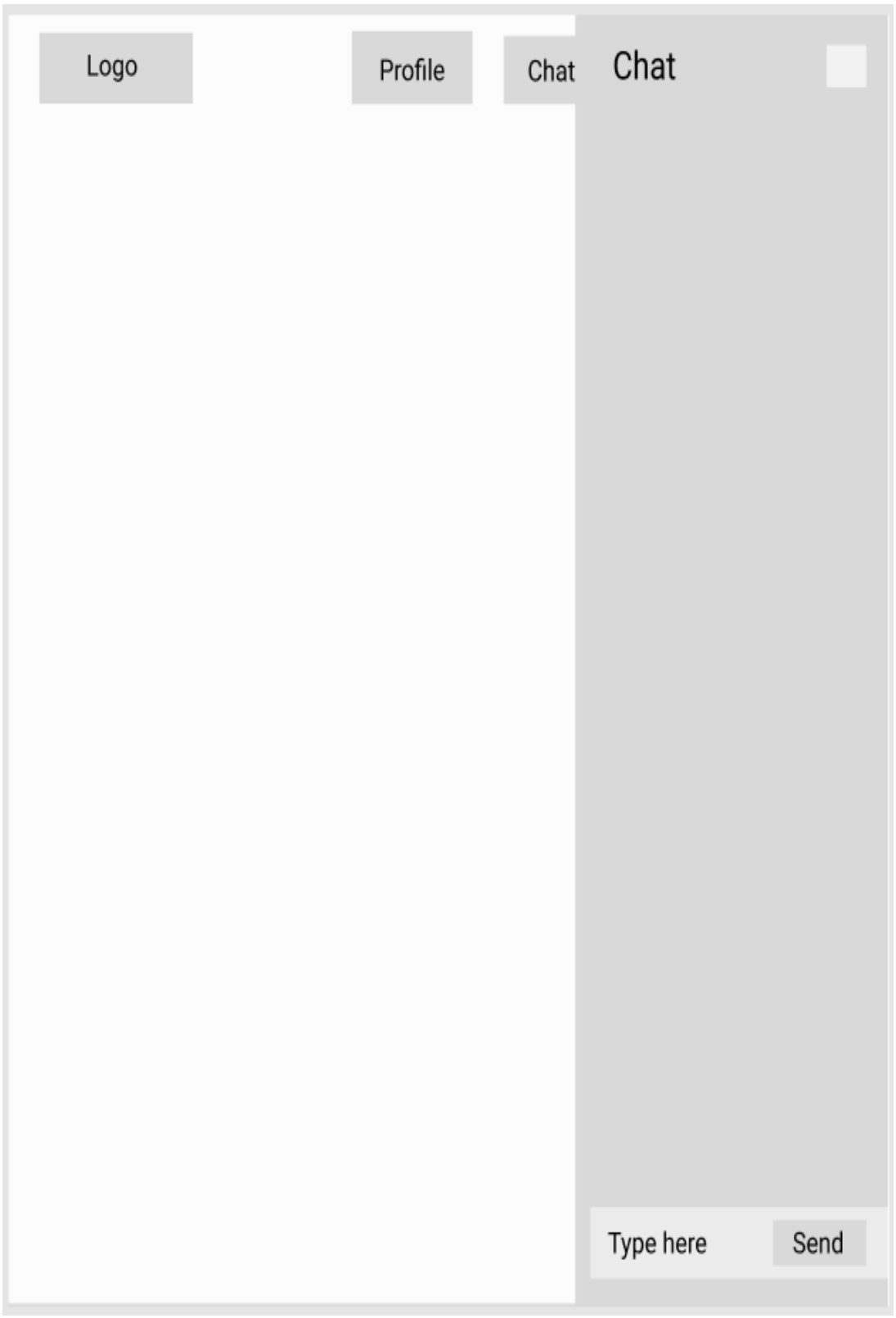

Logo
Login
Sign in using your google account
Continue with Google

Game

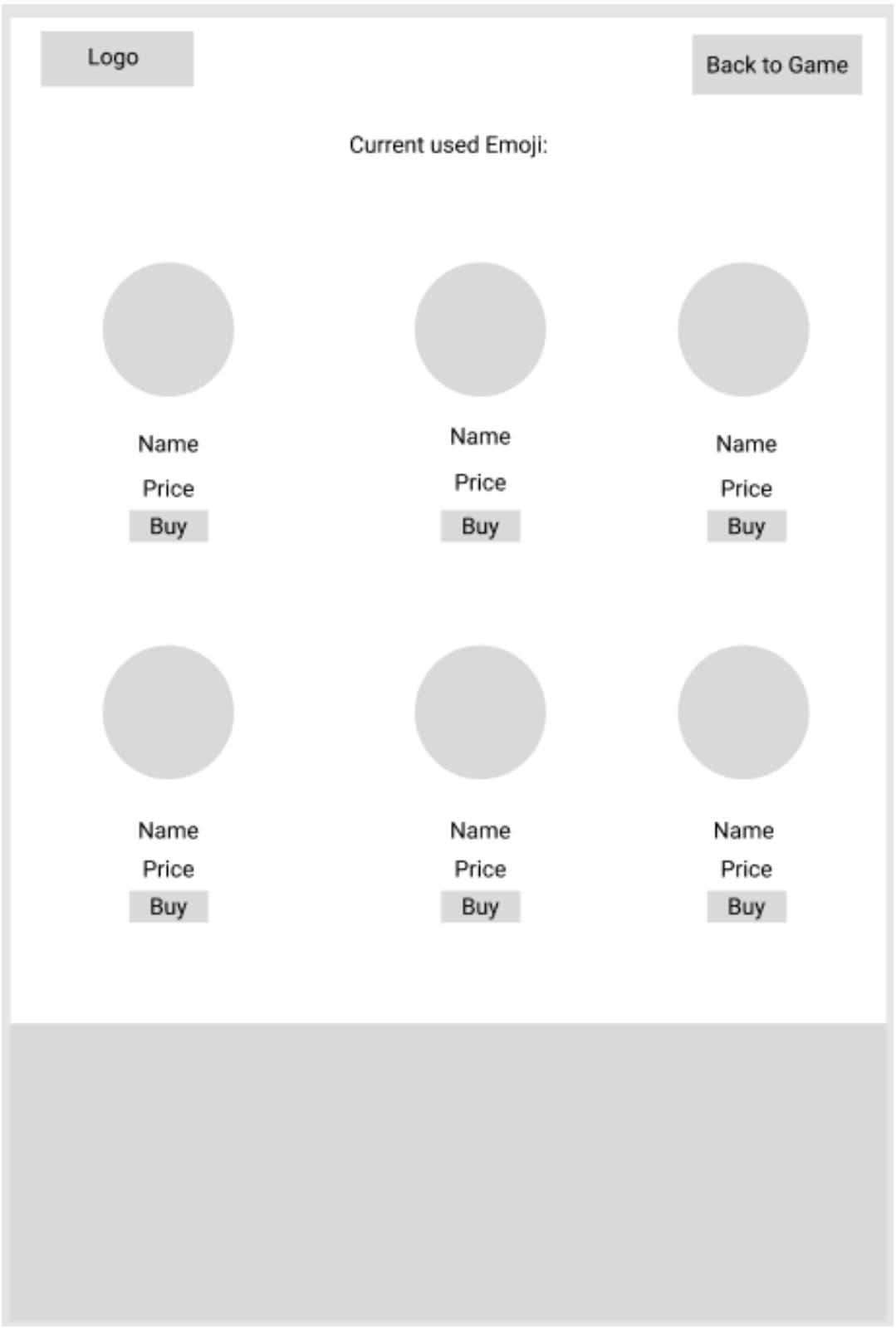
Profile


Logo
profile
Chat
shop
Logout
Profile
Name
Delete Account
Cancel

Chat



Shop


Logo
Back to Game
Current used Emoji:
Name
Price
Buy
Name
Price
Buy
Name
Price
Buy
Name
Price
Buy
Name
Price
Buy
Name
Price
Buy

Containerization

FunTime was containerised using Docker. To run FunTime locally with Docker, visit our DockerHub Repository.

Open Docker Desktop on your local machine. In your terminal, paste the following code:

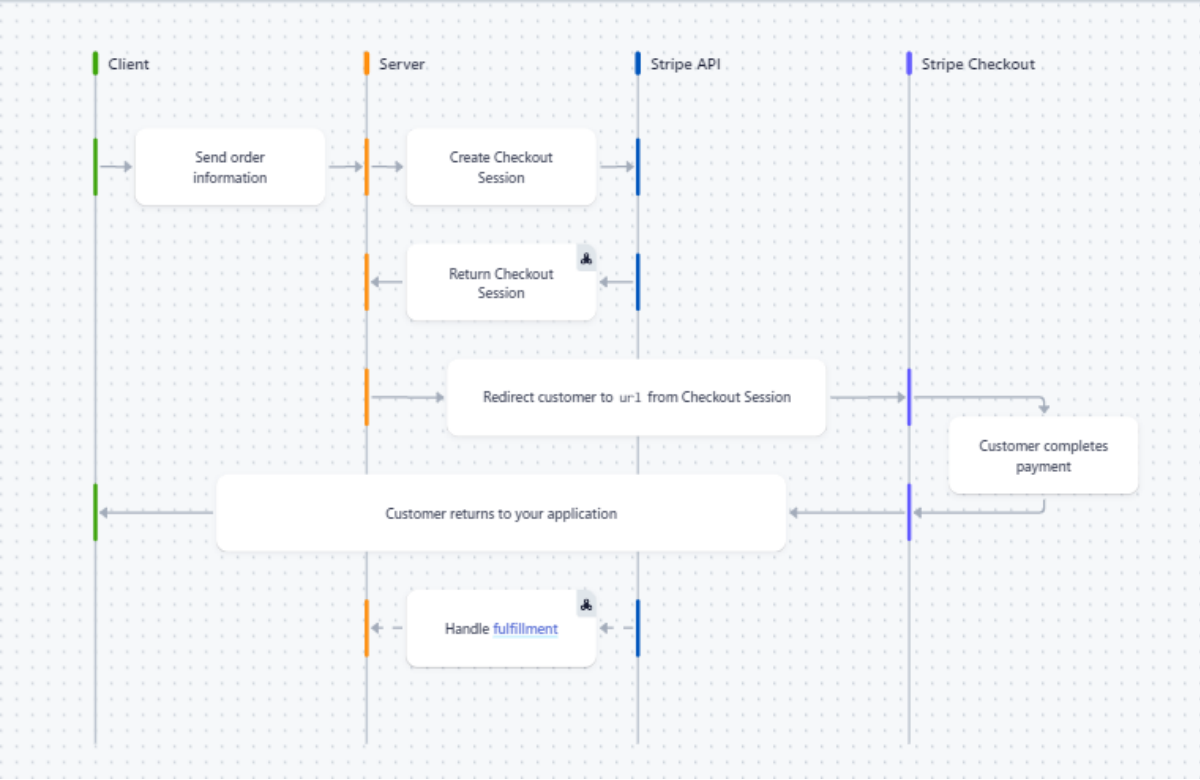
```
docker pull stephenasuncion/funtime:main
docker container run --name web -p 8080:8080 stephenasuncion/funtime:main
```

Internationalization

Presently, our app is only understandable or useful for people who understand English as there is no translation feature while chatting or for other elements. So, in future we will be considering implementing localization to allow translation of website to user local language and a feature to translate chat. We are using internationally accepted emojis and our documentation is updated to only include features that are available.

Diagrams

Payment data flow



source: https://stripe.com/docs/payments/checkout/how-checkout-works

API Documentation

Funtime's API documentation was generated from Postman. Full report can be found here.