



Happy Messaging

I'm Beside You Inc. Intern Task

-Harsh Verma, IIT Kanpur

Table of contents

01 Introduction

03 Dev Tools

02 Implementation

04 Services



1

Introduction

Happy Messaging revolutionizes the digital landscape by seamlessly uniting individuals across vast distances through lightning-fast real-time messaging and robust group chat capabilities, fortified by an ironclad email authentication protocol.

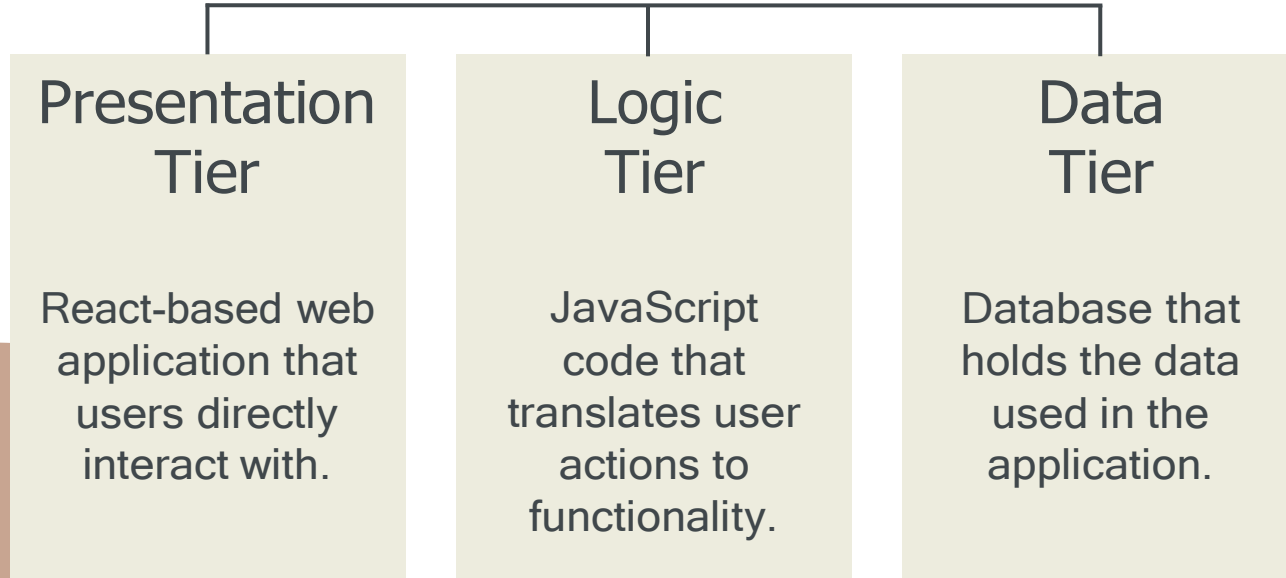




02

Implementation

Three Tier Architecture



Frontend

React JS

JavaScript
framework that
makes the website
dynamic

Tailwind

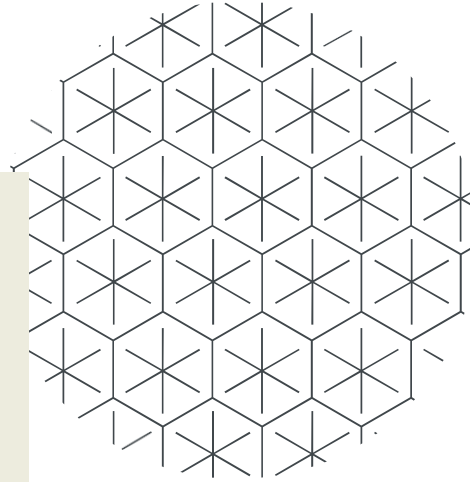
Design system that
helps maintain
consistency across
various design
requirements.

Backend

Backend is implemented in **Node.js and express.**

Node.js is used to create server-side web applications.

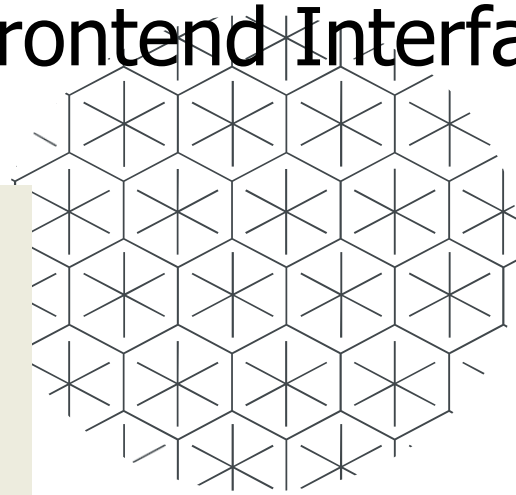
Express is very popular NodeJS framework that allows us to easily create APIs and backend.



Database & Backend -Frontend Interface

I have used **MongoDB Atlas**, is a multi-cloud database service. Database is hosted on **AWS (viaHeroku)**.

Socket.IO is a library that enables real-time, bidirectional, and event-based communication between the browser and the server.





03

Development Tools

Development & Version Control Environment



Git

Used as my
version control
system

GitHub

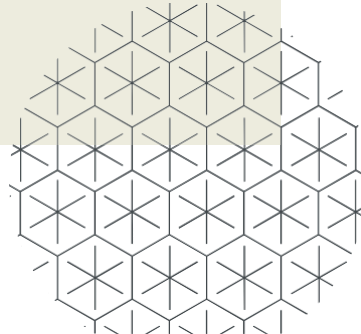
Used to manage
my repositories
and collaboration

CI/CD Pipeline

CI/CD is a method to frequently deliver apps to customers by introducing automation into the stages of app development.

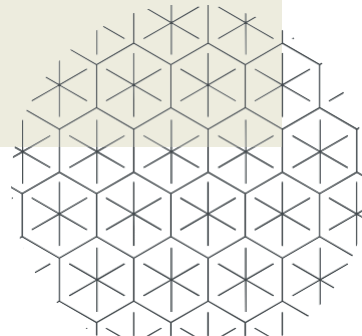
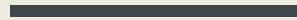
Development: Added actions that run on every pull request and detect any possible error and codesmells.

Deployment: Set up actions on my repo that continuously deploy the main branch of my applications' repo to the Heroku cloud.



04

Services



Hosting Service

Used **Heroku**, a platform as a service (PaaS), enables developers to build, run and operate applications entirely in the cloud.

All three tiers of my application, the frontend, the backend and the database run on the Heroku cloud.



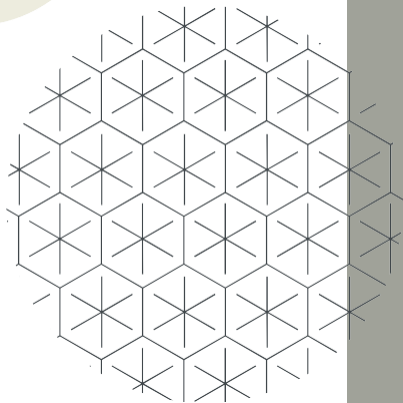
Authentication and Authorization

Authentication: stored the email and password in my database. The password uses **cryptographic** techniques of **hashing and salting** to prevent the data from being exposed in case of data leaks.

Authorization: I have made use of **JWT (JSON Web Token)**



Production Error Handling



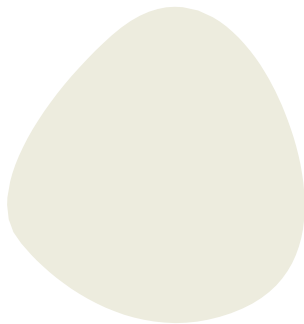
Sentry

I have used Sentry, a crash reporting platform to identify, investigate, and track the roadmap to fix bugs in production setup.

Future Development Plans

- Adding an online payment option.
- Deploying a corresponding mobile app.
- Customer feedback.
- Possibility of using other languages in the future.





Thank You

