SUMMER TRAINING REPORT
ON

"CHAT APP"

Submitted to
RAJASTHAN TECHNICAL UNIVERSITY

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

BY

VINAYAK MANGAL 16ESKCS182

UNDER THE GUIDANCE OF
MS. ANJANA SANGWAN

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SWAMI KESHVANAND INSTITUTE OF TECHNOLOGY , JAIPUR

2019-2020

Swami Keshvanand Institute of Technology, Jaipur

Department of Computer Science and Engineering

CERTIFICATE

Figure 1: Certificate

www.youstart.in

+91 - 8239089089

contact@youstart.in

INTERNSHIP CERTIFICATE

This is to certify that **Vinayak Mangal** pursuing B.Tech, 3rd year from SKIT, Jaipur has successfully completed "Full Stack Development Internship" at **YouStart**, Jaipur from **7th May 2019** to **1st July 2019**

He/She has worked on MERN stack and completed the project - **Multi-User Chat App**

We take this opportunity to thank him/her and wish all the best for his/her future.

FOR YOUSTART EDUCATION OPC PVT. LTD.

DIRECTOR

Abhishek Rathore
Director

YOUSTART EDUCATION (OPC) PRIVATE LIMITED
84, Kalyan Colony, Malviya Nagar
Jaipur, Rajasthan

CIN:U80302RJ2015OPC047194 Regd. Office: B-138, Ward No. 15, Agrasen Nagar, Churu - 331001, Rajasthan, INDIA

Acknowledgement

It is my pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behavior, and acts during the course of study.

I express my sincere gratitude to Mr. Mukesh Kumar Gupta Sir, HOD, CS/IT for providing me an opportunity to undergo summer training at YouStart Labs. I am thankful to Mr Abhishek Rathore for his support, cooperation, and motivation provided to me during the training for constant inspiration, presence and blessings. I also extend my sincere appreciation to Ms.Anjana Sangwan who provided his valuable suggestions and precious time in accomplishing my report.

At last I must express my sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped me directly or indirectly during this course of work.

VINAYAK MANGAL
16ESKCS182

Contents

List of Figures

Chapter 1

Introduction

1.1 OVERVIEW

Youstart Labs is Web Development and Training company that provides training in full stack development and Machine learning

Started in 2015, Youstart Labs has trained a large number of professionals and students across institutes and organizations in India. Our founding team is constituted of IIT, BITS Pilani alumni, Engineering Faculties, Scientists and Mentors from Top companies of the world.

1.2 MOTIVATION

JavaScript (JS) is one the most popular programming languages in the world right now, you can go and take a look at the Github Programming Ranking. Once supposed to a web front-end language now JavaScript run also on Backend servers, database servers, Mobile Front-ends and Hardware interfaces. JavaScript also making its marking world of hardware. Projects like Arduino and NodeBots have been showing that Javascript can become the primary language of embedded systems anytime now.

Youstart Labs is a young and energetic team of IIT – Delhi and BITS, Pilani alumni who are passionate about coding unlike software training institutes who lack such talent. We have worked as software developers and recruiters with big tech

companies such as Cisco, EMC Corporation, ISRO etc. and will provide you with a great insight into the skills required by such employer. We also run a different vertical as "software development team" where exceptional coders can work with us on real life projects. We have chosen the courses and content of the technologies after detailed discussions with our advisors, currently using these technologies at top software companies in India and abroad.

1.3 OBJECTIVES OF TRAINING

Youstart Summer Bootcamp is a blend of things - Learn Development Skills, Work on an Internship with other Organization, Interact with Mentors of Top Companies in the world, Build a Strong Online Resume and Profile for great career options.

Objective of my internship was to learn as much as I can and implement what i have learnt.Developing your knowledge of the structure and culture of a business and acquiring new skills.

Chapter 2

Introduction to Project / Modules

A very simple ,cross platform client server ”chat application” has been implemented using MERN STACK. The functionality of the chat application is to give the ability to chat with whoever is a registered user on the application.

2.1 OVERVIEW OF PROJECT/MODULES

In software architecture, there may be many layers between the hardware and end user. Each can be spoken of as having a front end and a back end. The front is an abstraction, simplifying the underlying component by providing a user-friendly interface, while the back usually handles business logic and data storage.

2.1.1 Front-end

Front-end web development is the practice of converting data to a graphical interface, through the use of HTML, CSS, and JavaScript, so that users can view and interact with that data.There are several tools and platforms (wordpress, magento etc..) available that can be used to develop the front end of a website, and understanding which tools are best fit for specific tasks marks the difference between developing a hacked site and a well designed, scalable site.
Front-end focus in this project is
1.Using Markup and web languages such as HTML, CSS JavaScript
2.Creating Single-page applications (with frameworks React)
3.Creating Responsive web design
4.Cross-browser compatibility issues and workarounds
5.Attractive User Interface

2.1.2 Back-end

The technology and programming that “power” a site—what your end user doesn’t see but what makes the site run—is called the back end. Consisting of the server, the database, and the server-side applications, it’s the behind-the-scenes functionality—the brain of a site. Back-end focus in this project is
1. Scripting languages used is Node.js
2.Scalability
3.Security concerns, authentication and authorization
4.Database administration and Management

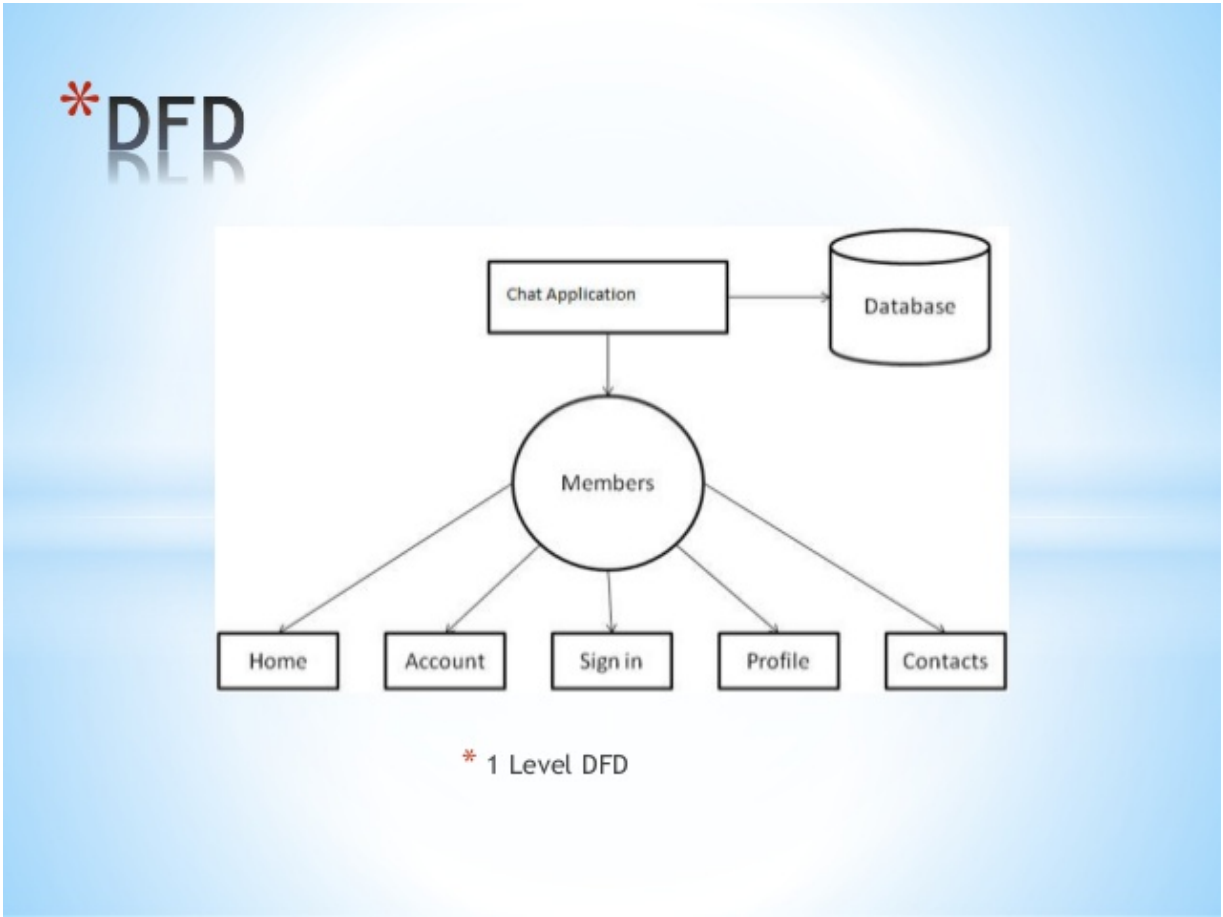
2.1.3 Deployment

Deployment of a Web application is similar to deployment of Connectors, EJBs, and Enterprise Applications. Like these deployment units, you can deploy a Web application in an exploded directory format or as an archive file.Web applications use two deployment descriptors to define their operational attributes. A web.xml deployment descriptor is a J2EE standard XML document that describes the contents of a WAR file. The weblogic.xml deployment descriptor is an XML document containing WebLogic Server-specific elements for Web applications.
The deployment plateform used for this app is ”Google FIREBASE”.

Chapter 3

Description of Modules

Figure 3.1: Data Flow Diagram



3.1 Front-End

Front-End consist of 3 pages-
1.Login page (Figure 4.1)
2.Sign up page (Figure 4.2)
3.User Dashboard (Figure 4.3)

Language Used : HTML,CSS,JAVASCRIPT.

Framework used : REACT JS

I had to create my front-end, and to make full use stack it would be best to use a JavaScript library for generating UI components. The MERN stack uses REACT js and thus I wanted to start with it. I used ReactJS a similar library by Facebook that is increasingly gaining popularity since its release. React was used to create the UI seen in the first figure. In comparison to Angular, which is a very complex framework, React is quite minimal because it is just plain JavaScript, whereas Angular leverages the benefits of templates. In my short experience with templates, I faced a situation in which template engines were extremely useful, but the React engine suffered greatly.
This is impossible with React by itself because it is translated to pure JS, which means I need to retrieve this data via Ajax. The problem is when I wanted to create a single-page application as seen in figure 1 where no html files are rendered, thus I am relying purely on React to manipulate the page, and I must use a different engine called Browserify in order to accomplish that and it does not provide this functionality. React itself is quite a unique library in its field. It has an object-oriented approach and builds UI by breaking them into classes and instances, and each components inherits from super classes similar to other OO languages. It also relies on methodologies used by game engines to update the UI in real-time.
React has its own virtual DOM loaded with the current state of the browser DOM. After an interaction, or an update to a component in React, a Diff operation between new state of the object and the one in the virtual DOM is called -similar to that of Git, which checks the shortest path to apply this change, and then patches the browser DOM. This is extremely important because native DOM operations are very slow, while the JavaScript implementation of the DOM used by React is superiorly faster, and this operation should be as efficient as possible.

Login and SignUP pages:

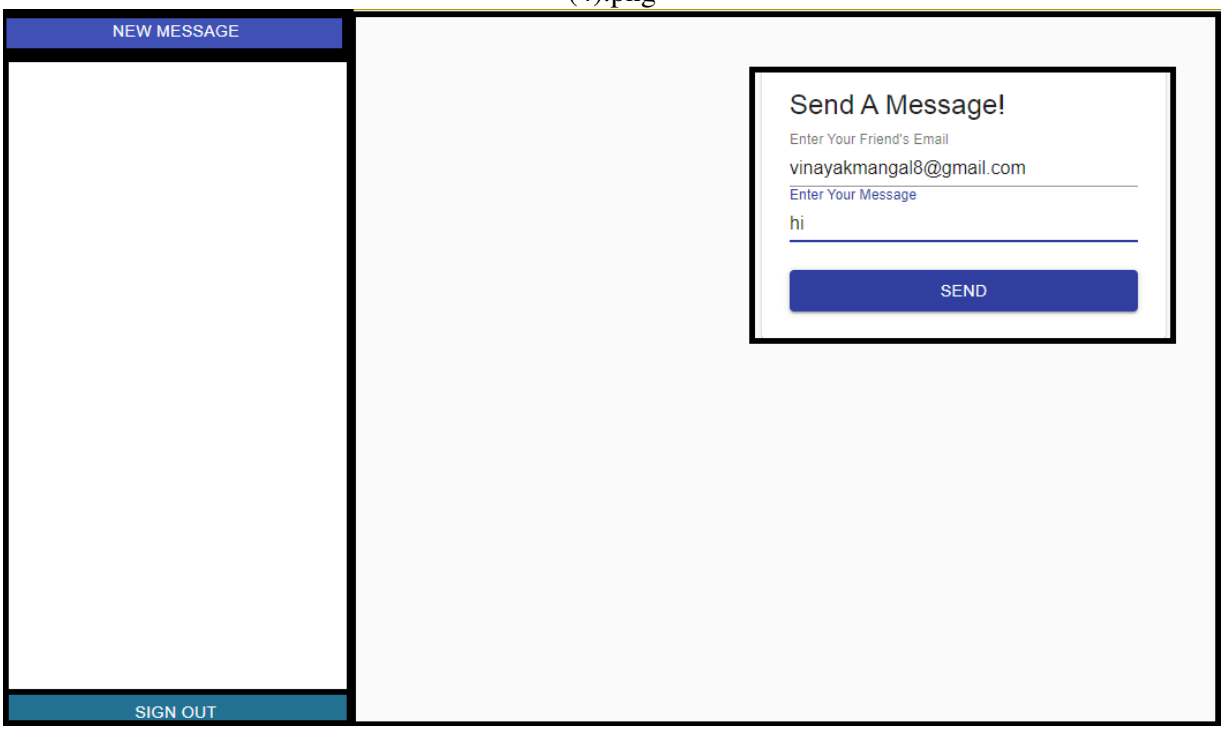
We need to create HTML form ,for collecting and submission of data from the user and send it to the server. So we use React js UI Framework which is ”Material UI”.Material-UI is an open-source project that features React components that implement Google’s Material Design.It kick-started in 2014, not long after React came out to the public, and has grown in popularity ever since. With over 35,000 stars on GitHub, Material-UI is one of the top user interface libraries for React out there.

Dashboard page:

Using React , I split each part of the main-page into a single React class components-in its own file. These components are Dashboard,Chat-view, Chat-text-box, New-chat, and Chat-list; all of which are imported in a main component class. Since React allows for easy management of state variables, I am able to easily send down data from the main component to the others. Nevertheless, sideways data communication is harder, for this reason I used Socket.io; by taking advantage of the fact that a single socket -each user has their own socket- can broadcast messages to itself, I can send events and data between single components.

Figure 3.2: Divided View of Dashboard page

(4).png



3.2 Back-End

Framework used : NODE JS,EXPRESS JS,GOOGLE FIREBASE

Node JS is a JavaScript platform for building fast, scalable, network applications built on Google's V8 Engine. Node is single threaded and built around the paradigm of none-blocking IO. With Node.js each incoming request by the user is handled by one single thread in opposition to the multi-threaded techniques used by PHP to scale the operations. Each request handled by this thread is coupled with a callback function that is called upon completion of the task. This is possible due to the fundamental support of JavaScript for events, Asynchronous operations, and callbacks; and Node.js puts JavaScript on the server side.

NPM

The Node Package Manager is based on JavaScript's npm. A built-in module that supports package management, it can be used to easily download and install modules for a Node application. Moreover, Node already has many packages and libraries developed to work on top of it; all of which confine to the asynchronous nature of Node.js.

Server

As Node can build an HTTP server out of the box, it can communicate with all other components through HTTP methods for CRUD operations based on the RESTful paradigm.

Although Node is capable of independently act as a web server, there are frameworks designed to make it more powerful and efficient -the most popular being Express.js. Express makes it possible to build an HTTP server very easily by wrapping the backend code of Node.js, and it makes the building of a RESTful API very simple.Routes are Express's way to handle incoming requests on a certain URL. Express after detecting the incoming requests directs it to a specific routing JavaScript file that handles all the logic associated with that URL. This is precisely helpful to keep the code lean and organized.

Database

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our Android, iOS, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.
The Realtime Database provides a declarative rules language that allows you to define how your data should be structured, how it should be indexed, and when your data can be read from and written to. By default, read and write access to your database is restricted so only authenticated users can read or write data. To get started without setting up Authentication, you can configure your rules for public access. This does make your database open to anyone, even people not using your app, so be sure to restrict your database again when you set up authentication.

Authentication

Most apps need to know the identity of a user. Knowing a user's identity allows an app to securely save user data in the cloud and provide the same personalized experience across all of the user's devices. Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

Firebase Authentication integrates tightly with other Firebase services, and it leverages industry standards like OAuth 2.0 and OpenID Connect, so it can be easily integrated with your custom backend.

To sign a user into your app, you first get authentication credentials from the user. These credentials can be the user's email address and password, or an OAuth token from a federated identity provider. Then, you pass these credentials to the Firebase Authentication SDK.After a successful sign in, you can access the user's basic profile information, and you can control the user's access to data stored in other Firebase products. You can also use the provided authentication token to verify the identity of users in your own backend services.

3.3 Deployment/Hosting

Firebase Hosting provides fast and secure hosting for your web app, static and dynamic content, and microservices.

Firebase Hosting is production-grade web content hosting for developers. With a single command, you can quickly deploy web apps and serve both static and dynamic content to a global CDN (content delivery network). You can also pair Firebase Hosting with Cloud Functions or Cloud Run to build and host microservices on Firebase.

Firebase Hosting is built for the modern web developer. Websites and apps are more powerful than ever with the rise of front-end JavaScript frameworks like Angular and static generator tools like Jekyll. Whether you are deploying a simple app landing page or a complex Progressive Web App (PWA), Hosting gives you the infrastructure, features, and tooling tailored to deploying and managing websites and apps.

Using the Firebase CLI, you deploy files from local directories on your computer to your Hosting server. Beyond serving static content, you can use Cloud Functions for Firebase or Cloud Run to serve dynamic content and host microservices on your sites. All content is served over an SSL connection from the closest edge server on our global CDN.

Firebase Hosting has lightweight hosting configuration options for you to build sophisticated PWAs. You can easily rewrite URLs for client-side routing or set up custom headers.

Implementation path:
1.Install the Firebase CLI
2.Set up a project directory
3.Deploy your site

Chapter 4

Assignments/Results

4.1 RESULTS ACHIEVED

We have successfully completed this project ,as a result we created the desired app.

4.1.1 Front End

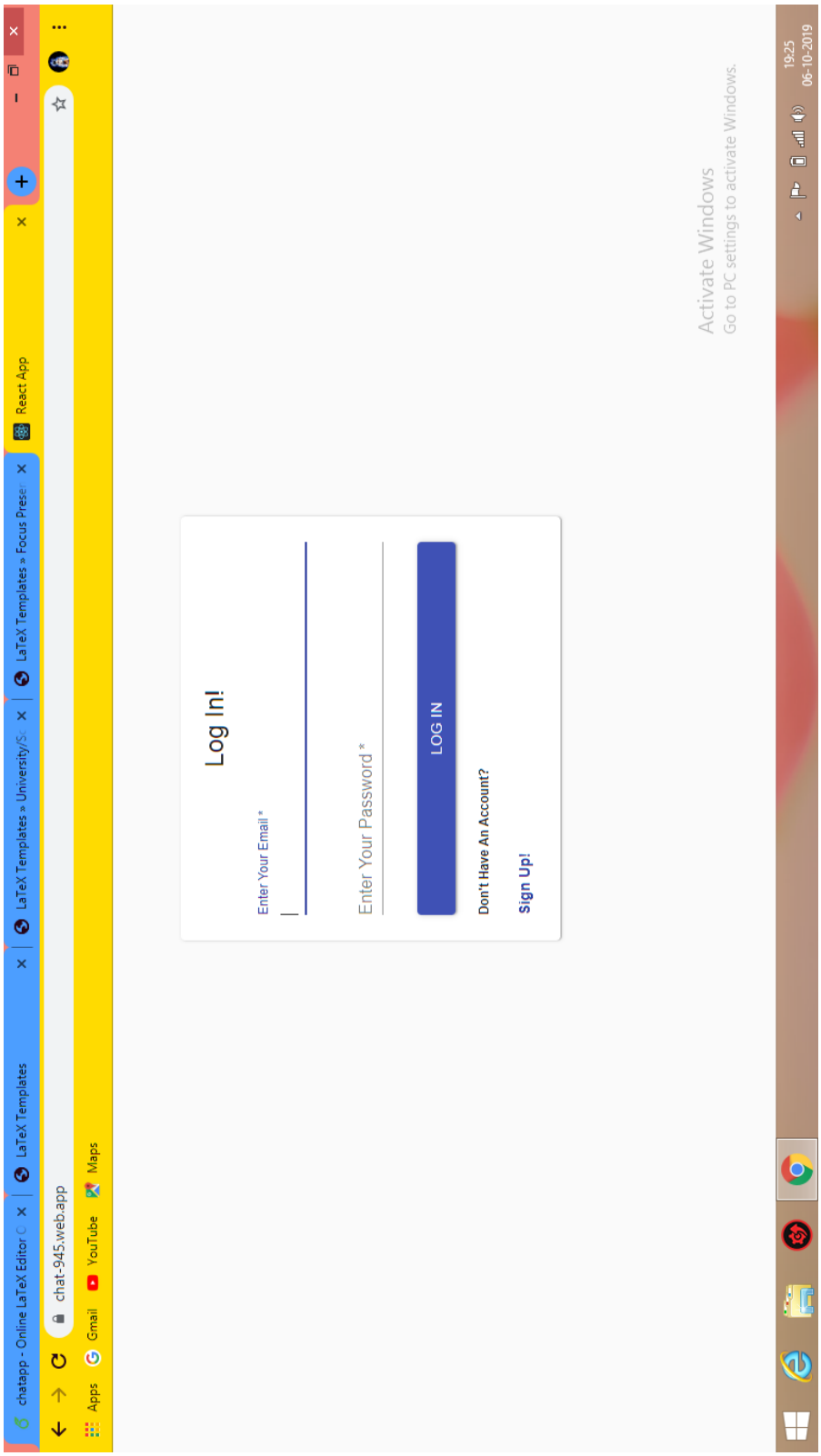
We successfully designed the layout of the app.The html and css used are working and we are able to observe change by changing it with the javascript.

4.1.2 Back End

We created firebase database and store our data there.The backend was working as expected.

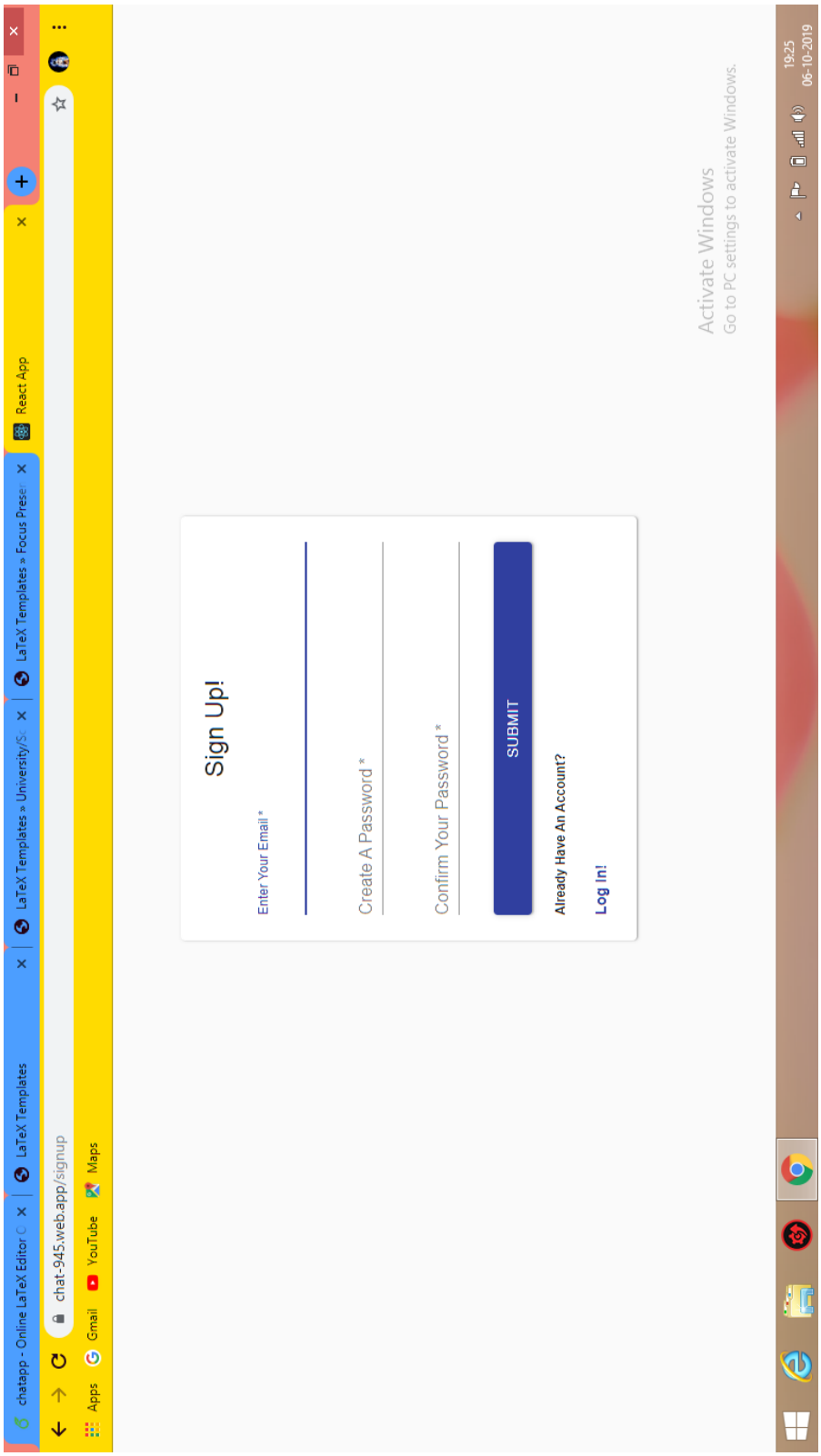
4.2 SCREENSHOTS OF ASSIGNMENT/RESULT ACHIEVED

Figure 4.1: Login Page



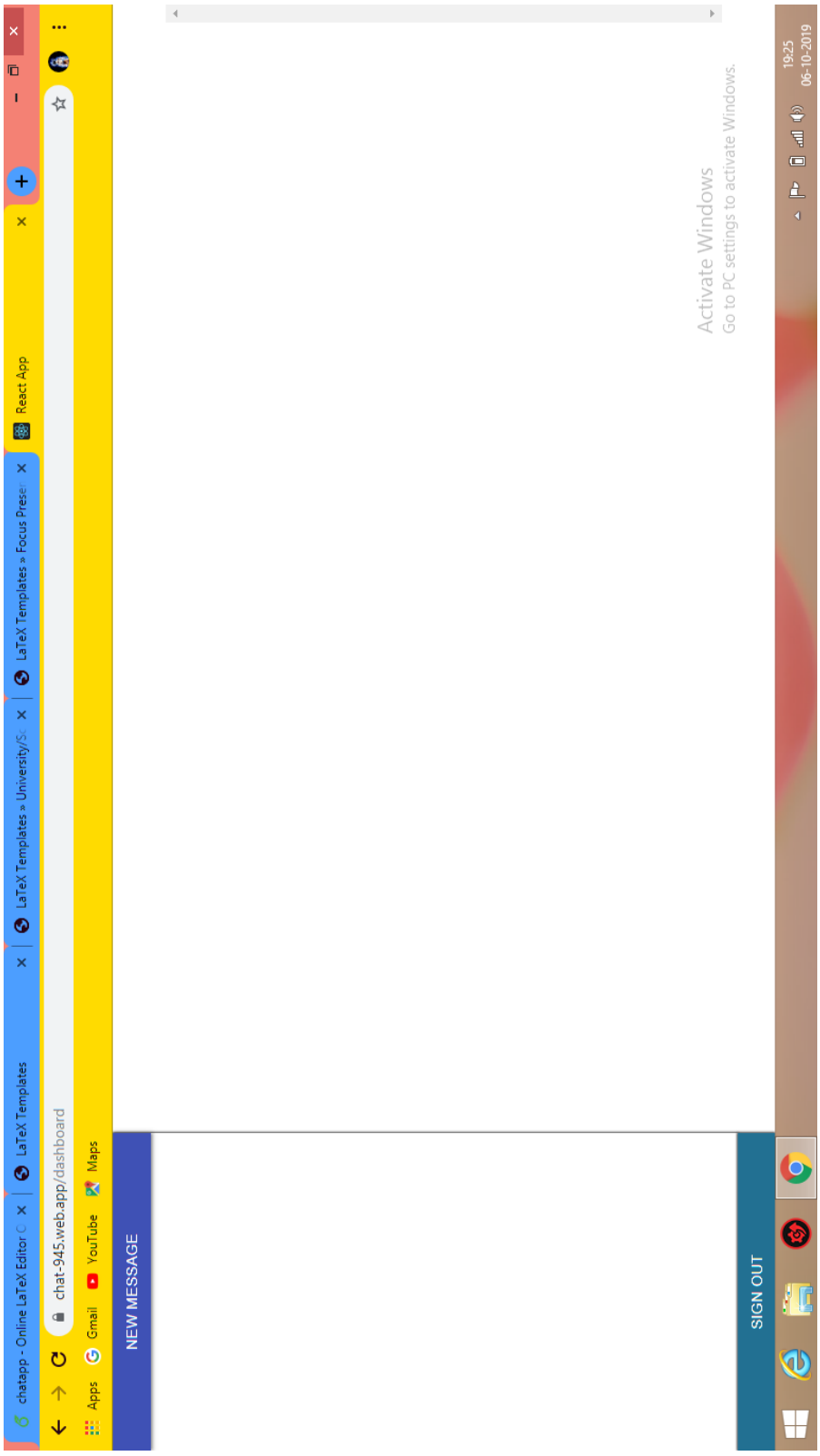
login.png

Figure 4.2: Sign UP page



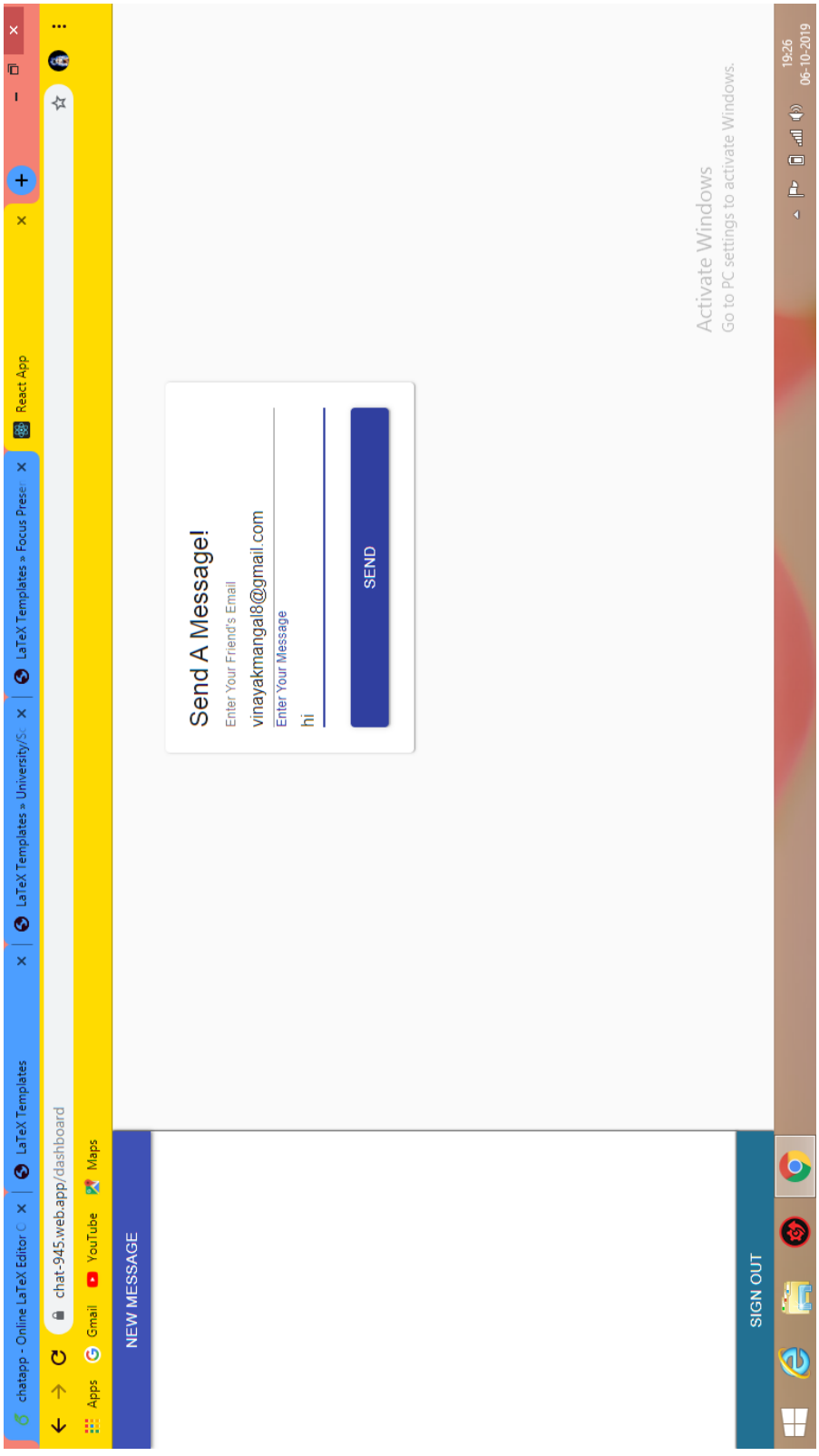
signup.png

Figure 4.3: Dashboard page



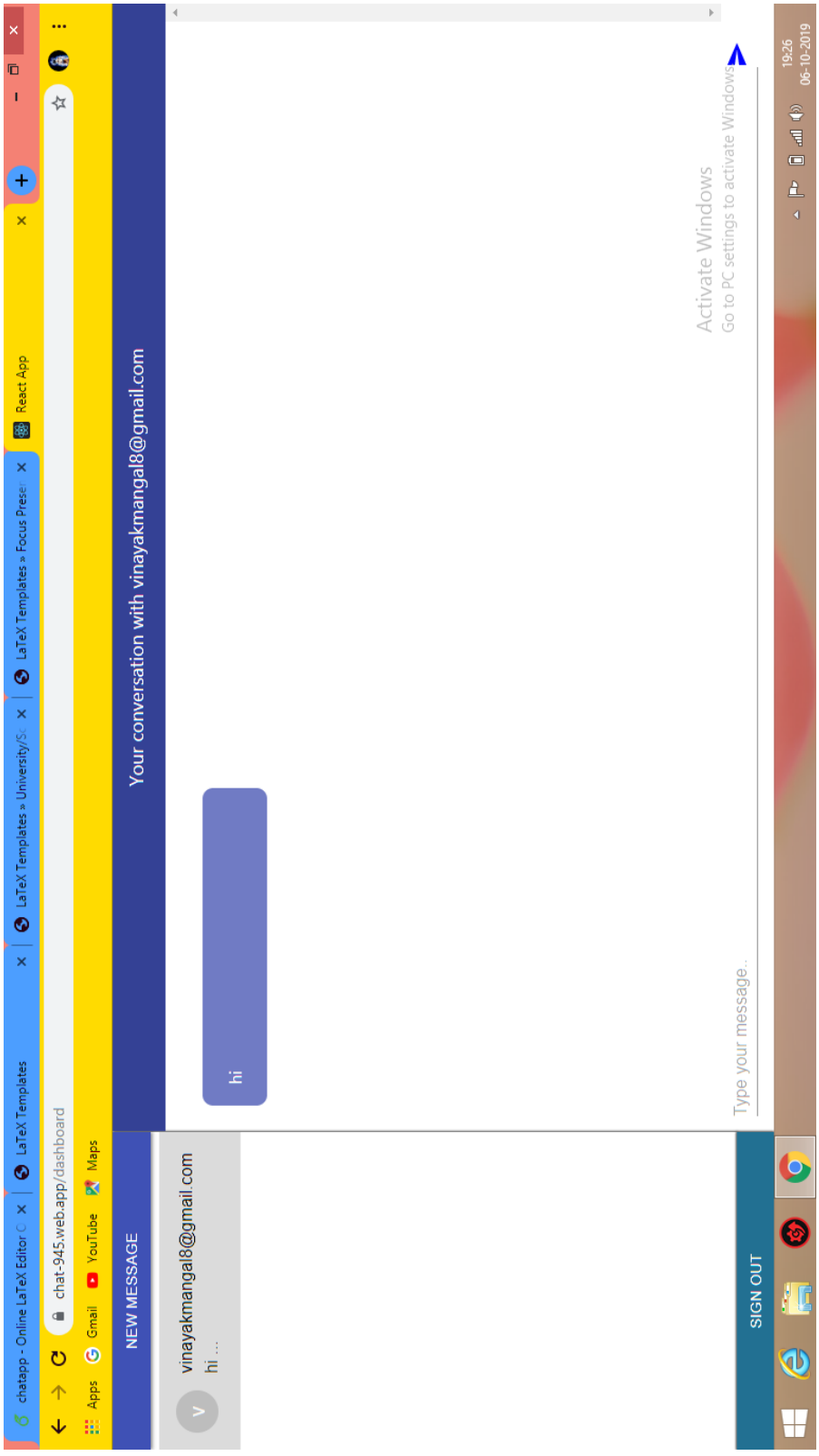
dashboard.png

Figure 4.4: New Message page



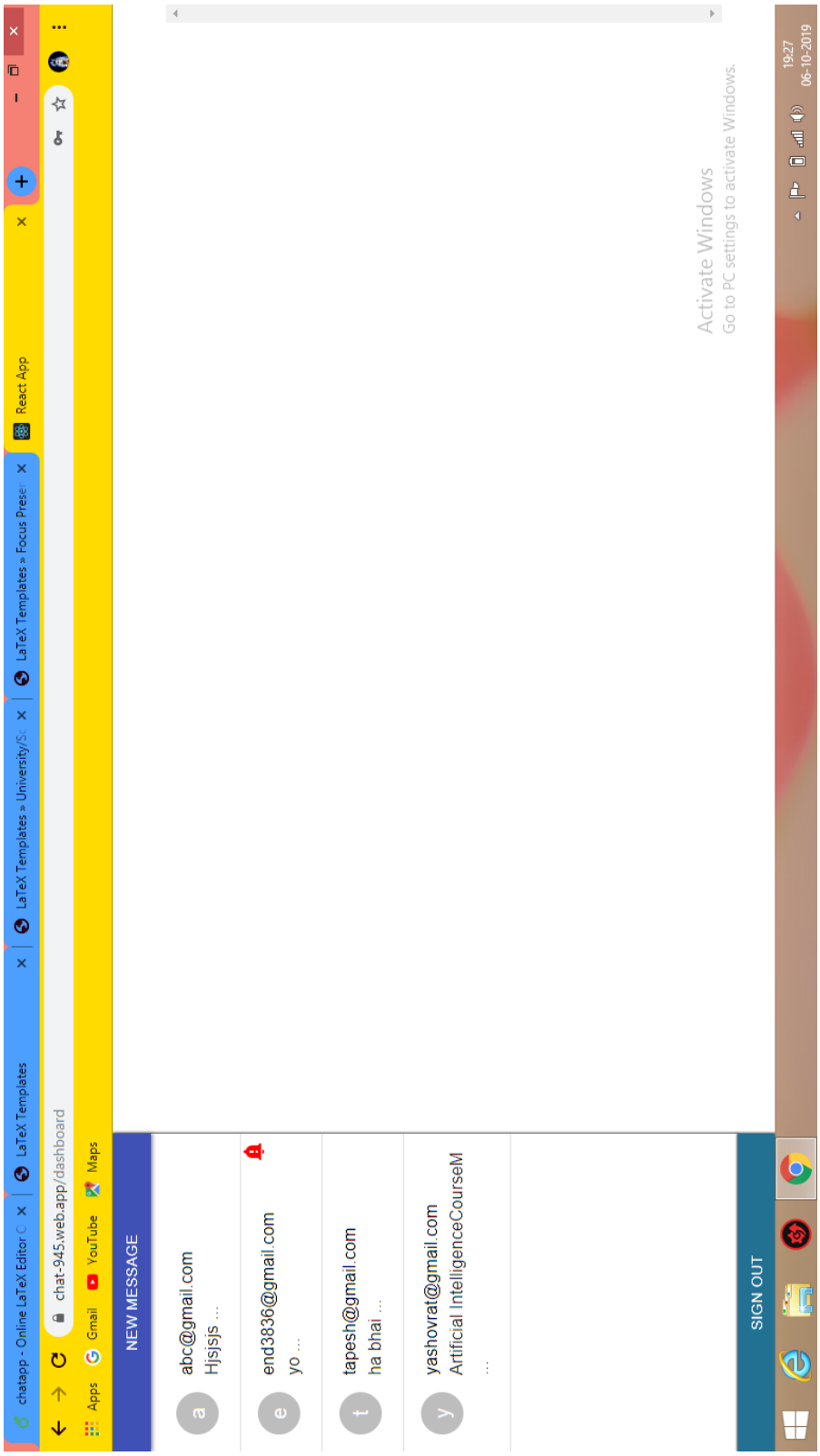
message .png

Figure 4.5: Chat send



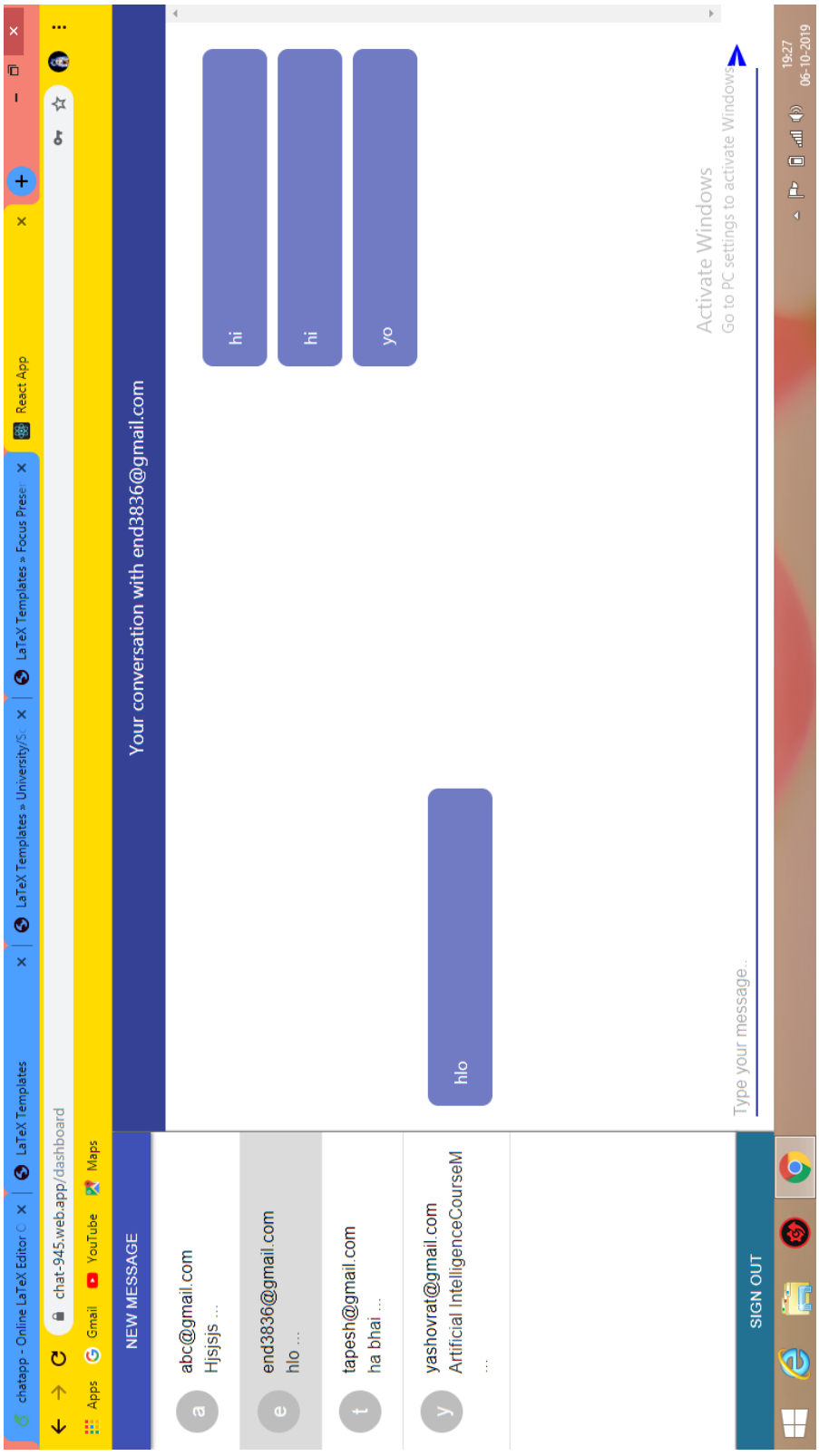
send.png

Figure 4.6: chat notification



notification.png

Figure 4.7: chat received and send



recieved.png

Chapter 5

Conclusion

5.1 TAKEAWAYS OF TRAINING

In this training period I was not only introduced to the number of technology but I was also introduced to a very friendly and healthy environment of the organization. My knowledge got increased in many fields. I got so much to learn about the new technology and also got an idea about how the IT industry runs,which technology is there in trend and which technology have a very nice scope in future.

5.2 FUTURE SCOPE

Chat app is a web application based on react that allows user to message to another user .
Data Record to empower real-time know of per-user-data usage analytics.
App Inbilt Store to allow for seamless downloading of apps/games from an inbuilt repository of third-party apps.
Attendance Management System, Leave Management System Will ensure that the entire office work is achieved on a single platform without shuffling across software. This will ensure saving in time and boost employee productivity.