

# **INTERNSHIP REPORT ON ADAM CHATBOT EXTENSION**

**(Feb 2022-June 2022)**

Internship report submitted in partial fulfillment of the requirement for the  
degree of Bachelor of Technology

in

**Computer Science and Engineering**

By

Bhawna Dey

181231



Department of Computer Science & Engineering and Information Technology

**Jaypee University of Information Technology,**

**Waknaghat, 173234,**

**Himachal Pradesh, INDIA**

## **DECLARATION**

I hereby declare that this submission is my own work carried out at Paxcom India (P) Ltd - A Paymentus Company, Delhi from February 2022 to June 2022 and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text.

**Submitted by:**

**Bhawna Dey**

**181231**

Computer Science & Engineering and Information Technology Department

Jaypee University of Information Technology

## **ACKNOWLEDGEMENT**

I would like to thank Madhuri Sankhyan, Manager of HR in Talent Acquisition, of Paxcom India (P) Ltd - A Paymentus Company, Delhi for allowing me to do an internship within the organization.

I also would like to thank Mrs. Pallavi Vasishta and all the people that worked along with me at Paxcom India (P) Ltd - A Paymentus Company, Delhi with their patience and openness they created an enjoyable working environment.

I also would like to thank Mr. Sunil Garg for mentoring me throughout my internship.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I am highly indebted to Mr. Pankaj Kumar, Training & Placement Coordinator of our college for the facilities provided to accomplish this internship. I would also like to thank the Head of our Department Dr.Vivek Kumar Sehgal and the faculty for teaching us the skills required for this internship.

Finally, I must acknowledge the constant support and patients of my parents.

Bhawna Dey

181231

Jaypee University of Information Technology

# TABLE OF CONTENT

<b>Content</b>	<b>Page No.</b>
<b>Declaration By Candidate</b>	<b>I</b>
<b>Acknowledgement</b>	<b>II</b>
<b>List of Figures</b>	<b>IV</b>
<b>Chapter 01: INTRODUCTION</b>	<b>1</b>
1.1 Introduction	1
1.2 Job Description	1
1.3 Objectives	2
1.4 Internship Schedule	2
<b>Chapter 02: COMPANY DESCRIPTION</b>	<b>4</b>
2.1 Paxcom	4
2.2 Paymentus	5
<b>Chapter 03: TOOLS AND TECHNOLOGIES USED</b>	<b>7</b>
3.1 Github	7
3.2 Postman	8
3.3 Typescript	10
3.4 MongoDB	11
3.5 Express.js	13
3.6 React	14
3.7 Node.js	17
<b>Chapter 04: LIVE PROJECT</b>	<b>19</b>
4.1 Description	19
4.2 Method to Create API keys	19
4.3 Building and Deploying the Chrome Extension	20
4.4 Code	21
4.5 Results	24
<b>Chapter 05: CONCLUSION</b>	<b>31</b>
5.1 Conclusion	31
5.2 Mentor's Review	31
<b>REFERENCES</b>	<b>32</b>
<b>PLAGIARISM REPORT</b>	<b>33</b>

## LIST OF FIGURES

<b>Fig no.</b>	<b>Figures</b>	<b>Page no.</b>
1	Objectives of Software Engineering	1
2	About Paxcom	4
3	About Paymentus	6
4	My Github Project Repository	8
5	Postman Application	10
6	Example of TypeScript Class	11
7	An example of a MongoDB Application	13
8	Express Framework	13
9	An example of an Express.js application	14
10	An example of a ReactJS application	15
11	Features of ReactJS	6
12	An example of a Node.js Application	18
13	Login Page	24
14	Signup Page	25
15	Check mail Page	26
16	Home Page	27
17	Sentiment Page	28
18	Google Authorization	29
19	Facebook Authorization	29
20	Twitter Authorization	30

# Chapter 01:INTRODUCTION

## 1.1 Introduction

An internship is a professional teachable moment that gives students real-world experience in their field of study or professional goals. An internship enables students to learn new skills while exploring and enhancing their careers. This report is a description of my ongoing internship at Paxcom India (P) Ltd - A Paymentus Company, Delhi. This internship report details the activities that helped me achieve a bunch of my stated objectives. I was assigned the profile of “MERN Stack Developer” for my internship. We were asked to learn about the languages in the first three months of the internship which was followed by one month of independent projects assigned by the mentor. After the completion of this learning process, I was assigned the live project of the company name “Adam Chatbot”. I was asked to develop and handle the social signup part of the chrome extension for the Adam Chatbot.

## 1.2 Job Description

The introduction of engineering concepts to the creation, implementation, and technical management is known as software engineering. Software engineering was designed to address the challenges of low-quality software initiatives. Problems arise when software exceeds schedules, budgets, and quality expectations. It assures that the software is built in a consistent, correct, timely, cost-effective, and specification-compliant manner. Software engineering became vital to keep up with the rapid depending of user needs and the context in which the program is expected to operate.

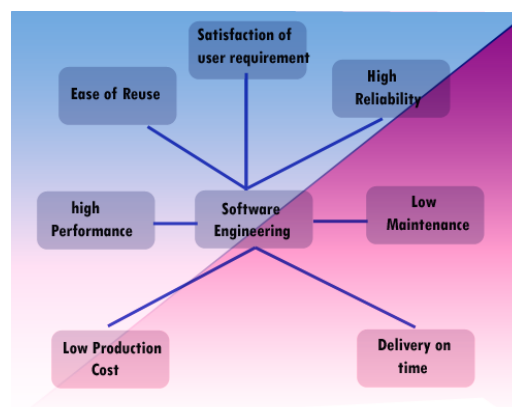


Fig 1. Objectives of Software Engineering

The Software Engineer Trainee is responsible to assist in the design and development of software. The Software Engineer Trainee works with other members of the team to develop secure and dependable software solutions. The duties and responsibilities of this position are as follows:

- Application development (coding, programming)
- Code debugging and testing
- New software application documentation and testing
- Investigating, diagnosing, and resolving a wide range of technical challenges
- Working with senior executives
- Identifying issues and developing solutions
- Learning about new technology ahead of time

### **1.3 Objectives**

The objective of the internship is to learn more about the programming languages like Node.js, Express.js, MongoDB, and ReactJS. These four languages are the basic requirements for my job profile. After the learning process, I was assigned to work on a live project for the company. This internship provides experience to the freshers so that they can learn more about the industry. The perspective was to gain enough knowledge so that we can work easily on the project assigned to us.

### **1.4 Internship Schedule**

The internship plan was as follows:

➤ February 2022

This month, I revised the HTML, CSS, and Javascript which was followed by learning about Node.js. I was assigned to develop a simple node.js application for hands-on experience and a better understanding of the language.

➤ March 2022

To learn more about backend development, I learned about Express.js and MongoDB in this month. I developed a backend project using all three languages. This project helped me to

know how the backend works for the web development project.

➤ April 2022

After the completion of backend development, I studied ReactJS for frontend development. On learning all the languages for MERN Stack development, I was assigned to create an NGO application to integrate all my learnings into one project. This project helped me to understand how the web application works. Later this month, I was assigned the live project of the company.

➤ May 2022

I worked on creating social media signup of Google, Facebook, and Twitter for the Adam ChatBot extension so that the user can signup directly.

➤ June 2022

We are planning to work further on this project and add the features required by the client.



# Chapter 02: COMPANY DESCRIPTION

## 2.1 Paxcom

We're a group of 400+ eCommerce enthusiasts that are enthusiastic about using technology and our experience to make digital commerce easier for brands in global marketplaces.

Some of the world's leading brands such as PepsiCo, Mondelez, Britannia, Lenovo, Wipro, and Abbott trust our data, analytics, consulting, and solutions to inform and execute their e-commerce-driven retail strategy. Paxcom is a subsidiary of Paymentus, a global leader in transparent digital billing and payment solutions.



Fig 2. About Paxcom

### 2.1.1 Our Services

- **Channel Intelligence**-You'll get on-demand access to full data with revenue, competitiveness, category, and promotion information thanks to our dynamic KPI dashboards. Our patented technology can gather and analyze publically available data across many platforms and geographies, allowing us to make better informed, qualified judgments.
- **Strategy and Execution**-Our team of eCommerce professionals, data and business

analysts, article writers, and graphic professionals can assist you with pricing policy, marketing, and ranking on top eCommerce sites.

- **Campaign and AMS Management-** For your brand, our highly devoted staff can plan, implement, analyze, and optimize searching and advertising (CPC and CPM) campaigns. We have a lot of experience with Amazon AMS promos, Flipkart PLA advertisements, and other resources that can assist your customers with their purchasing decisions.
- **Omni Channel Solutions-**We simplify warehouse operations and order fulfillment from beginning to end. To help you keep track of inventory over platforms, like eCommerce, local retailers, and distributors, our solution effortlessly interacts with major logistics services and marketplaces like Flipkart and Amazon.
- **Marketplace Seller Solutions-** We're here to help you improve order fulfillment, inventory tracking, and account reconciliation. Our superior digital capabilities can assist you in reconciling purchases and returns and identifying additional costs instantly — this is necessary for payment monitoring, consolidation, and dispute resolution.

## 2.2 Paymentus

Paymentus is a major cloud-based payment technology and solution provider. Over 1,700 clients and financial institutions receive our upcoming product suite via a modern technological stack. Millions of businesses and consumers in North America utilize our platform to interact with our clients. We processed about 2.5 billion monetary transactions in 2021.

We work with non-discretionary service providers of various sizes in a number of industries, including industries, business institutions, healthcare, administration, telecommunications, and healthcare. Our Instant Payment Network, or IPN, expands the reach of our platform. Our IPN partners give their customers access to the full possibilities of our next suite of products, as well as the ability to interact with and pay our big and expanding list of billers, by connecting to our IPN. As a result of these partnerships, our platform now reaches millions more customers in the US and around the globe.

### MORE WAYS TO PAY AND GET PAID WITH THE INSTANT PAYMENT NETWORK®

Our Instant Payment Network (IPN) extends the reach of our platform, giving businesses more ways to connect to their customers through their preferred channels, all with a single integration to Paymentus.

IPN makes it possible for consumers to manage and pay bills through Amazon Alexa, inside the PayPal app, at businesses like Walmart and via online banking for hundreds of banks.



Fig 3. About Paymentus

## Chapter 03: TOOLS AND TECHNOLOGIES USED

### 3.1 Github

Software developers and engineers can use GitHub to build public-facing cloud repositories for free. You can transfer a GitHub repository to your device, add and alter files locally, and then "push" your changes back to the repository, where they will be visible to the public.

#### Repository

A repository (sometimes shortened as "repo") is a site where all of the files for a project are kept. Each project has its repository, which you can access using a specific URL.

#### Forking a Repo

When you establish a new project based on an existing project, this is known as "forking." This is a fantastic tool that greatly fosters the development of new applications and projects. If you find a project on GitHub that you'd want to contribute to, you can fork it, make the modifications you want, and then republish it as a new repo. You may quickly add changes to your current fork if the original repository that you forked to create your new project is updated.

#### Git Commit

You're ready to commit once you've staged the files you want to add. The commit message is important regardless of whether you commit using GitHub Desktop or the command line. Short commit statements that describe your change should be used. The commit messages will lead you through the history of your repository, therefore they should be informative. The following message format can be used in command-line commits:

```
git commit -m "git commit message example"
```

#### Pull Requests

You've forked a repository, made a fantastic change to the project, and want the original devs to notice it—perhaps even put it in the original project/repository. Create a pull request to accomplish this. The original repository's authors can view your work and decide whether or not

to accept it into the official project. When you submit a pull request, GitHub provides an excellent communication channel between you and the primary project's maintainer.

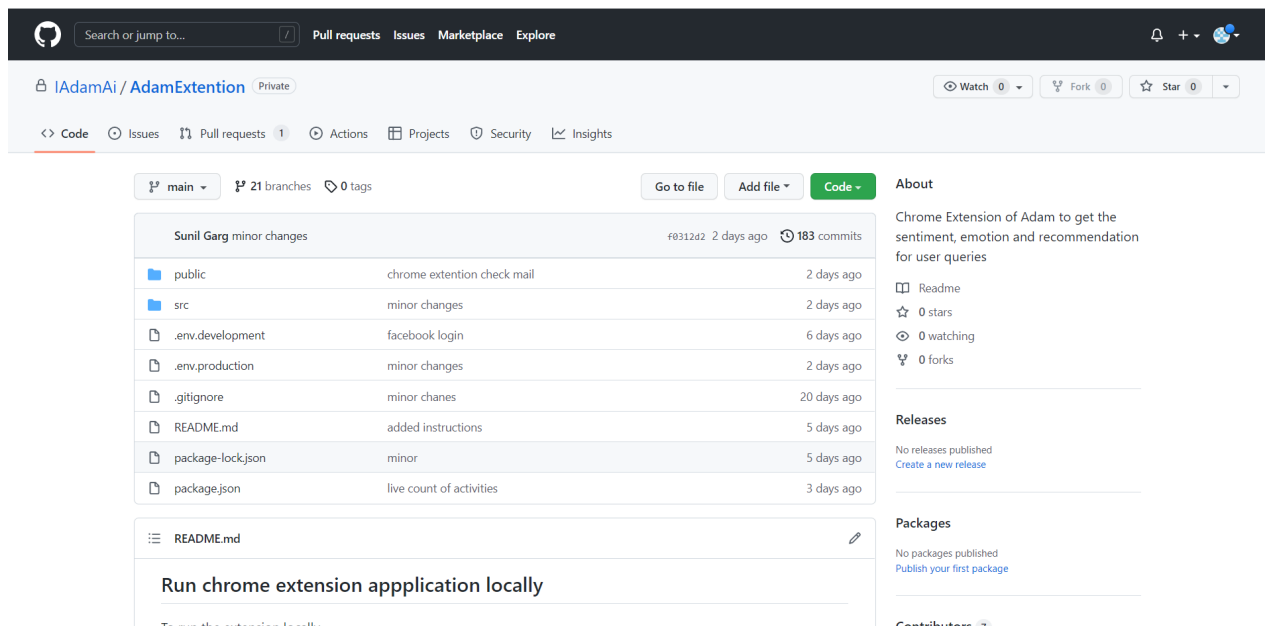


Fig 4. My Github Project Repository

### 3.2 Postman

Postman is a well-known API testing tool. APIs may be simply created, tested, shared, and documented with the help of this tool. Postman is an Application Program Interface (API) testing tool that may be used to create, test, develop, change, and document APIs. It's a basic graphical user interface for executing and analyzing HTTP requests and responses. When utilizing Postman for testing, you don't need to write any HTTP connection code. Instead, we use Postman to connect with the API and create test suites called collections. Almost every feature a developer can require is included in this program. This tool can do GET, POST, PUT, and PATCH HTTP queries, as well as convert APIs.

Postman is built around a set of powerful technologies that are extremely easy to use. Postman has become a useful tool for more than 8 million users. Postman is used for the following reasons:

1. Accessibility- After installing Postman on the device, simply login to the account to use it anywhere.

2. Use Collections- Users can create collections for their API calls using Postman. Each set can create multiple requests and subfolders. It will aid in the organization of test suites.
3. Test development- Every API request should include verification of a successful HTTP response status to test checkpoints.
4. Automation Testing- The Collection Runner or Newman can conduct tests in multiple repetitions or iterations, saving time for repeated tests.
5. Creating Environments- The usage of numerous environments reduces test replication because the same collection can be used in different settings.
6. Debugging- The postman console assists in successfully debugging the tests by tracking what data is being obtained.
7. Collaboration- To improve file sharing, you can import and export collections and environments. You can also share the collections via a direct connection.
8. Continuous integration- It can support continuous integration.

It's as simple as typing a URL into your browser to send a request. In Postman, we can quickly send requests to APIs. You can use an API request to access or deliver data from a data source.

An HTTP method is required to send the API request. POST, GET, DELETE, PUT, and PATCH is some of the most widely used methods.

- GET: This HTTP method is used to access the data from an API.
- POST: This method transmits new data.
- DELETE: This is used to remove or delete the existing data.
- PATCH: This method is used to update the existing data.
- PUT: This method is used to update the existing data.

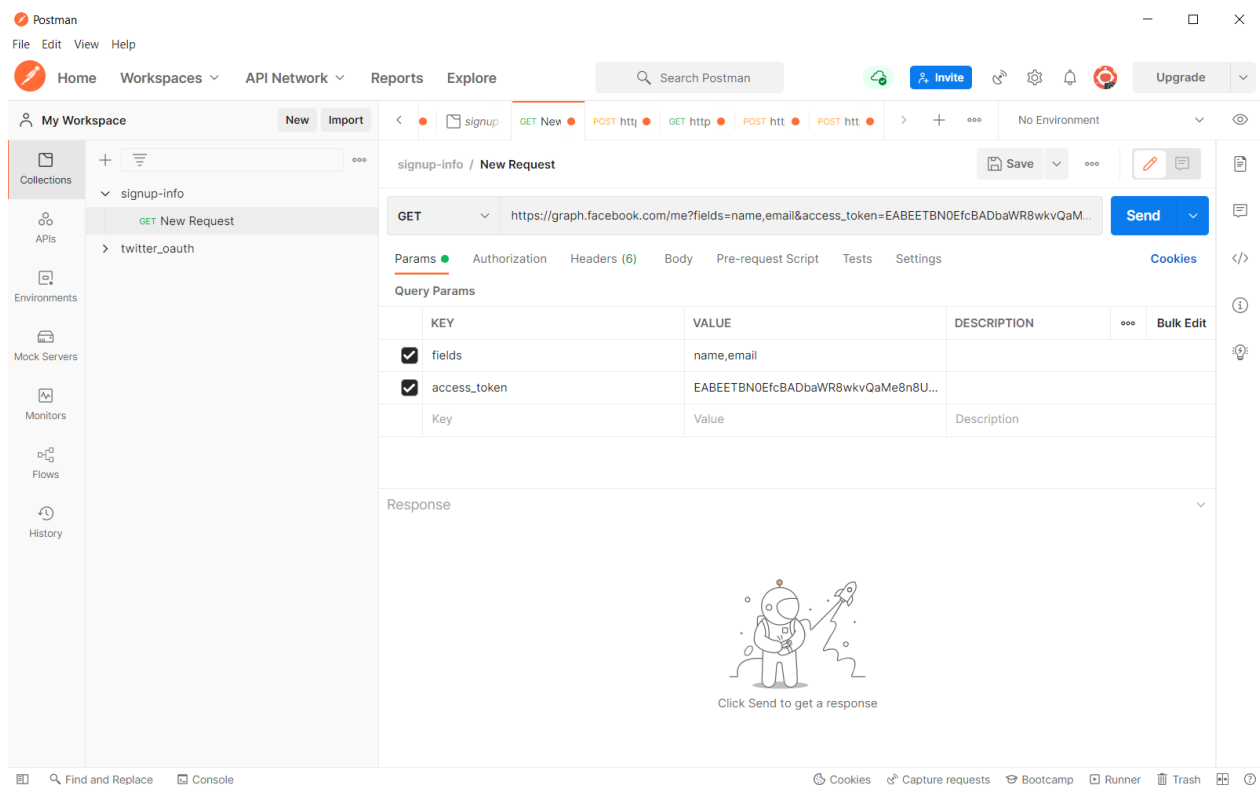


Fig 5. Postman Application

### 3.3 Typescript

TypeScript was invented and is maintained by Microsoft. It's a strict syntactic and semantic javascript framework with the feature of static typing as an option. It employs JavaScript and is designed for the construction of large apps. Because TypeScript is a javascript framework, existing JavaScript programs are also valid TypeScript programs.

Both server-side and client JavaScript applications can be built with TypeScript (as with Node.js). You can use the TypeScript Analyzer or the Babel compiler to translate TypeScript to JavaScript.

TypeScript includes definition files that describe the structure of current JavaScript libraries, similar to how C++ header files describe the architecture of existing object files. Variables defined in the files can be used by other programs as if they were TypeScript properties with statically typed values. Third-party header files are included in popular frameworks like jQuery, Mongo, and D3.js. TypeScript annotations for Node.js fundamental modules are also available,

allowing TypeScript to be used to write Node.js apps.

The TypeScript compiler is written in TypeScript and then compiled to JavaScript. Its use is governed by the Apache License 2.0. TypeScript is supported as a first-class scripting language with C# or other Microsoft programs in Visual Studio 2013 Version 2 and later. Visual Studio 2012 now supports TypeScript thanks to an official plugin. The TypeScript was developed by Anders Hejlsberg, the principal architect of C# and developer of Delphi and Turbo Pascal.

```
class Person {
  private name: string;
  private age: number;
  private salary: number;

  constructor(name: string, age: number, salary: number) {
    this.name = name;
    this.age = age;
    this.salary = salary;
  }

  toString(): string {
    return `${this.name} (${this.age}) (${this.salary})`; // As of version 1.4
  }
}
```

Fig 6. Example of TypeScript Class

### 3.4 MongoDB

MongoDB is a document-oriented, cross-platform database with exceptional productivity, availability, and easy handling. MongoDB is built on the collection and document concepts.

#### Database

A database is a tangible repository for data. On the file system, each database has its own collection of files. Typically, a MongoDB server provides many databases.

#### Collection

In MongoDB, a collection is a group of documents. An RDBMS table is the same thing. A collection exists within a single database. Collections do not enforce a schema. Within a collection, different elements can be found in separate documents. In most cases, every one of the documents in a collection has the same or similar functions.



## **Document**

A document is made up of a set of key-value pairs. Documents have a dynamic schema. Documents in the same collection don't really want to have the same number of fields or architecture, and similar variables in a collection's document can contain data.

### **3.4.1 Advantages of MongoDB over RDBMS:**

- MongoDB is a way to store data that allows you to store multiple documents in one collection. A document's amount of features, content, and size may differ from one to the other.
- The structure of a single thing is clear.
- No complicated connections here.
- Deep querying capabilities. MongoDB supports dynamic querying on documents using a document-based markup language that's also nearly as effective as SQL.
- Tuning.
- The scalability of MongoDB is simple.
- No conversion or mapping of application components to database objects is required.
- The (windowed) work set is stored in internal memory, allowing for quicker data access.

### **3.4.2 Uses of MongoDB:**

- Data is saved as JSON-style documents in document-oriented storage.
- Indexing can be done on any attribute.
- Replication and high availability.
- Auto-Sharding
- Rich searches
- For real-time information.
- Expert assistance is provided by MongoDB.

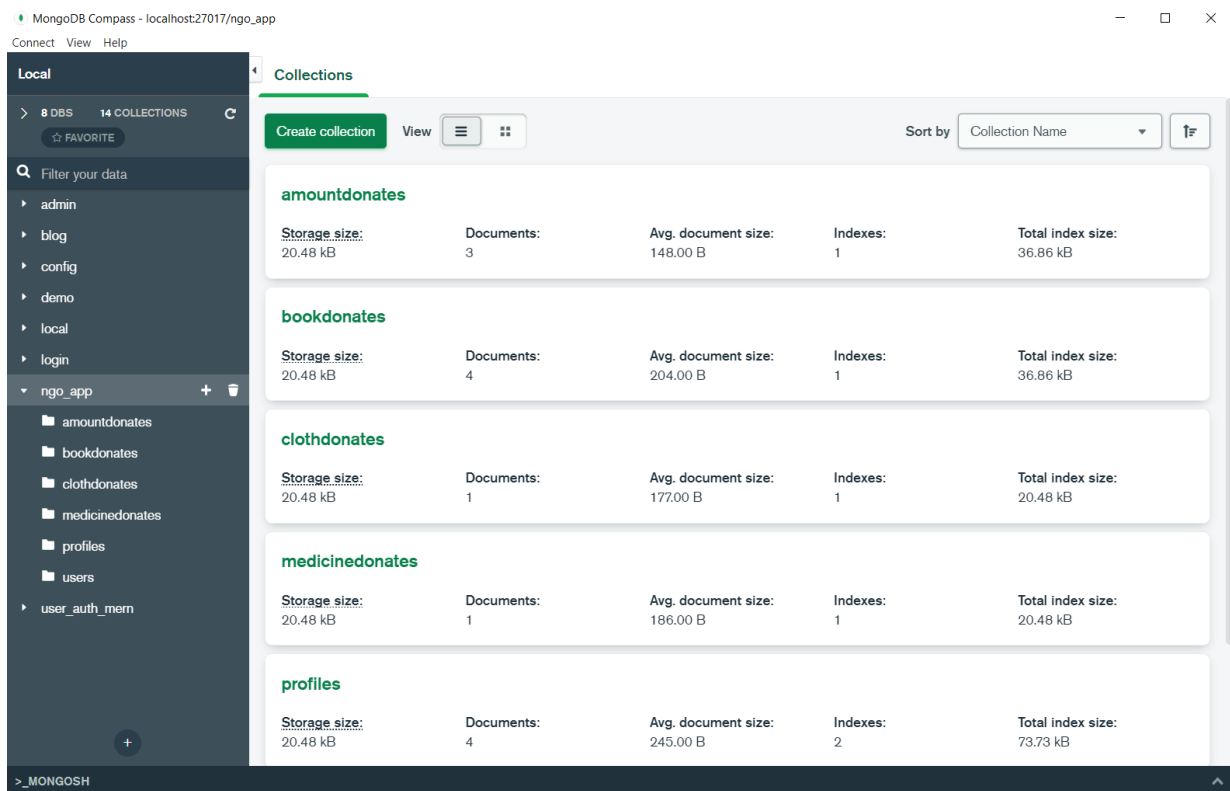


Fig 7. An example of a MongoDB Application

### 3.5 Express.js

It is a Node.js web framework that includes a wide range of features for developing web and mobile apps. It makes it easy to swiftly build Node-based Web apps. The following are some of the important aspects of the Express framework:

- Allows middlewares to build up responses to HTTP queries.
- Creates a routing table depending on the HTTP Method and Address to perform various actions.
- Allows you to dynamically generate HTML pages using template arguments.



Fig 8. Express Framework

### 3.5.1 Advantages of Express.js

1. Quickly and easily creates Node.js web applications.
2. Configuration and customization are simple.
3. You could use HTTP methods and URLs to define application routes.
4. There are various middleware modules included that can be used to carry out extra requests and responses.
5. Several template engines, such as Jade, Vash, EJS, and others, are easy to use.
6. Allows you to choose a middleware for error handling.
7. It's simple to provide static files and resources from your app.
8. Creates a REST API server for you.
9. Connecting to databases like MongoDB, Redis, and MySQL is simple.

```
1  const express = require('express' 4.18.1 )
2  const app = express()
3  const port = 3000
4
5  app.get('/', (req, res) => {
6    res.send('Hello World!')
7  })
8
9  app.listen(port, () => {
10   console.log(`Example app listening on port ${port}`)
11 })
```

Fig 9. An example of an Express.js application

### 3.6 React

ReactJS is a Javascript framework for constructing declarative, fast, and adaptable user interface components. It's a component-based front-end framework that manages the app's view layer exclusively. Jordan Walke, a software engineer at Facebook, conceived the idea. Facebook invented and maintains it, and it's later used in Facebook products like WhatsApp and Instagram. Although ReactJS was designed around 2011 for Facebook's feed area, it was first publicly published in May 2013.

The most common website architecture today is MVC (model view controller). React is the view in the MVC design, whereas Flux or Redux is the architecture.

A ReactJS application is made up of numerous components, each being responsible for generating a reusable piece of HTML code. Components are the foundation of all React programs. Complex applications can be developed using simple building blocks by nesting these components with other components. ReactJS interconnects the DOM-based technique for populating data in the HTML DOM. Because the virtual DOM alters specific DOM components rather than refreshing the full DOM, it is fast.

To create a React app, we create React components that correspond to various aspects. The application structure is made up of these components, which are arranged into higher-level components. Consider a form containing the required fields, labels, and buttons, among other things. Each form component can be created as a React element, which we then combine to build the form component. The form elements would describe the structure and parts of the form.

ReactJS' major purpose is to construct a User Interface that aids program performance. It makes use of a virtual DOM (JavaScript objects), that speeds up the app. In JavaScript, the virtual DOM is quicker than the traditional DOM. ReactJS can also be used on server and client sides, as well as with other frameworks. Component and data structures are used to frame and make app maintenance easier.

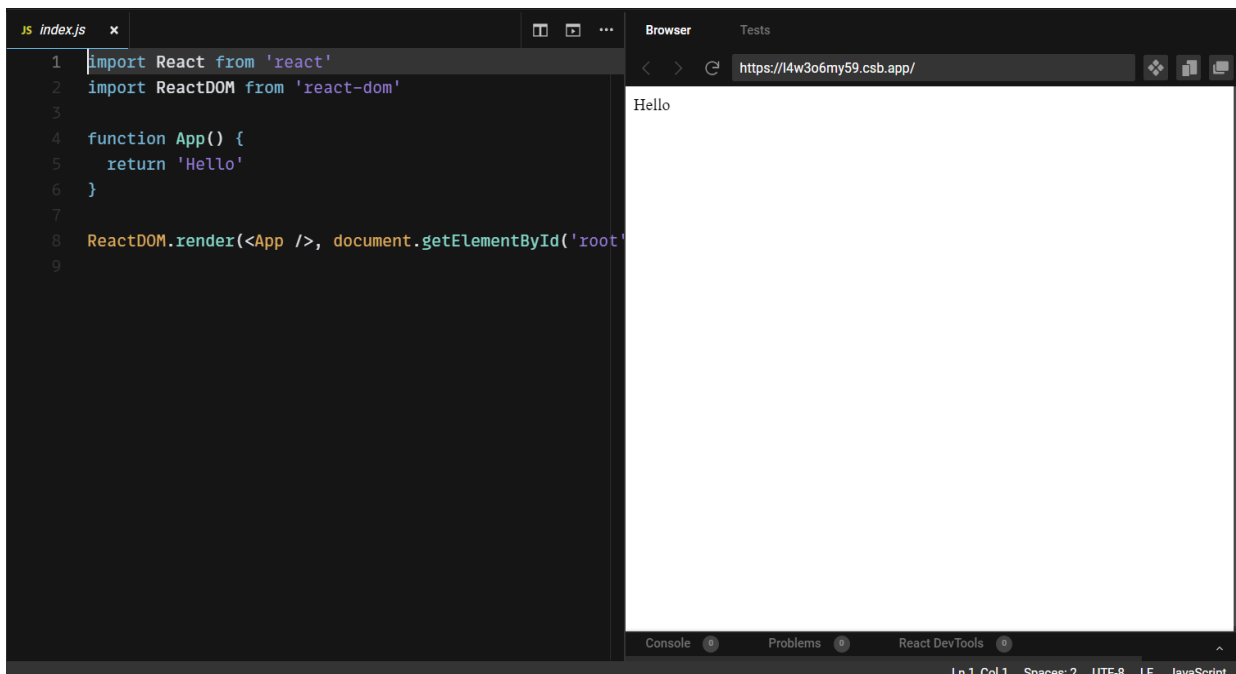


Fig 10. An example of a ReactJS application

### 3.6.1 Features of ReactJs

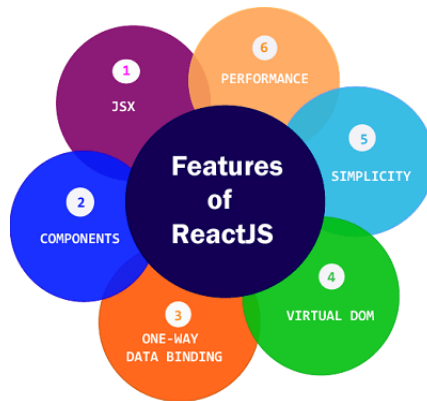


Fig 11. Features of ReactJS

ReactJS is quickly becoming the most popular JavaScript framework among web developers. It is an important part of the front-end ecology. The following are some of ReactJS' key features:

- JSX

JavaScript XML is abbreviated as JSX. It's a syntax extension for JavaScript. ReactJS uses an XML or HTML-like syntax. This syntax is converted into React Framework JavaScript calls. It enhances ES6 to allow HTML-like text and JavaScript react code to coexist. It is not required to utilize JSX, however, it is strongly recommended in ReactJS.

- Components

Components are at the heart of ReactJS. A ReactJS application consists of several components, each with its logic and controls. These components are reusable, which makes it easier to keep the code clean while working on larger projects.

- One-way Data Binding

ReactJS is designed to handle one-way data binding or unidirectional data flow. One-way data-binding allows you to have more control over your application. If the data flow occurs oppositely, additional characteristics are required. Because components are designed to be immutable, and the data they contain cannot be modified, this is the case. Flux is a pattern that

aids in data unidirectionality. This increases efficiency by making the application more versatile.

- Virtual DOM

The original DOM object is represented by a virtual DOM object. It functions similarly to a one-way data binding. The complete UI is re-rendered in virtual DOM representation whenever any changes are made to the web application. Then it compares the differences between the old and new DOM representations. After that, the true DOM will only update the things that have changed. This speeds up the application and eliminates memory waste.

- Simplicity

ReactJS makes use of a JSX file, which makes the program easy to code and comprehend. We already know that ReactJS is a component-based solution that allows you to reuse code as needed. This makes it easy to use and understand.

- Performance

The speed of ReactJS is well-known. This distinguishes it from other frameworks currently available. This is due to the fact that it manages a virtual DOM. The Document Object Model (DOM) is a cross-platform computer programming API for HTML, XML, and XHTML. The DOM is purely memory-based. As a result, we didn't even write directly to the DOM while creating a component. Instead, we'll create virtual components that will be converted into the DOM, resulting in smoother and faster performance.

### **3.7 Node.js**

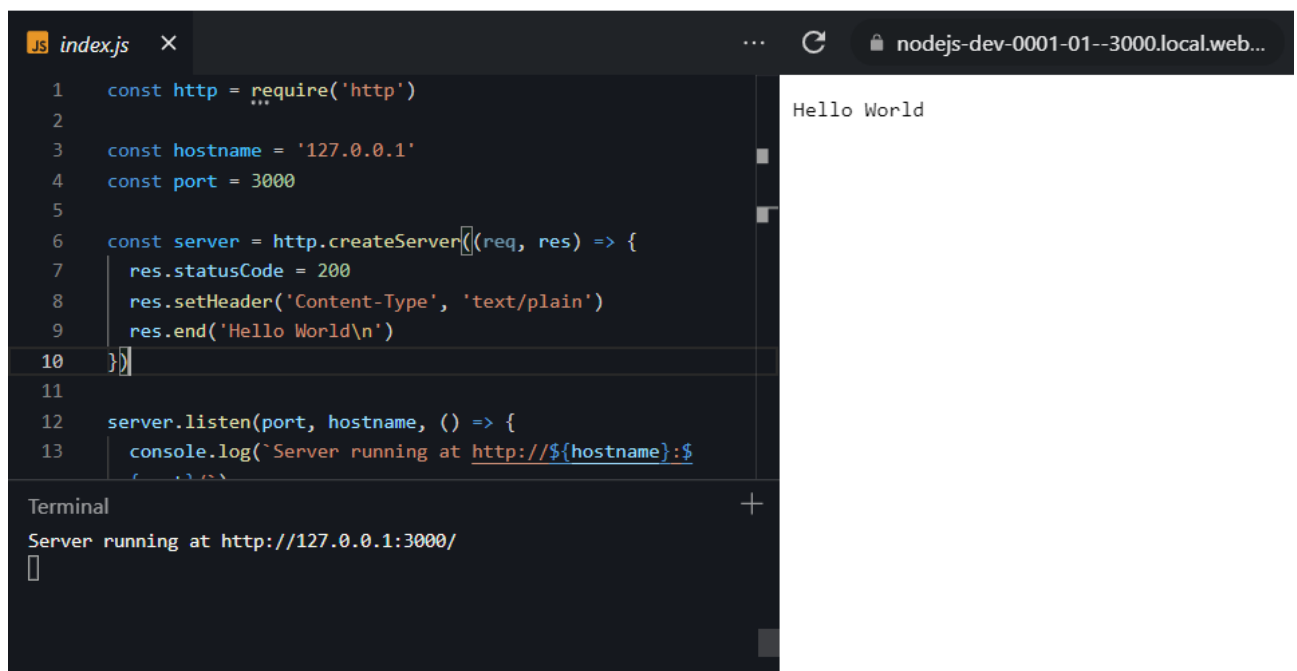
Node.js is an open-source and cross-platform JavaScript runtime environment. It is a commonly used tool for almost any task. Node.js runs the V8 JavaScript engines, which is at the core of Google Chrome, outside of the browser. As a consequence, Node.js is lightning quick.

Node.js software runs in a single operation rather than creating a new process for each request. The standard library of Node.js offers a set of asynchronous I/O semantics that prohibit JavaScript code from stopping, and modules in Node.js are frequently built using non-blocking

paradigms, thus blocking behavior is an exception to the general rule.

When Node.js performs an I/O operation, such as receiving from the connection, entering a database, or entering the filesystem, instead of suspending the process and spending CPU cycles awaiting a response, Node.js will begin the activities as soon as the response is received. This allows Node.js to handle hundreds of concurrent connections without incurring the cost of thread concurrency control, which may be a major source of mistakes.

Node.js has a significant benefit since billions of frontend programmers who develop JavaScript for the internet can now generate server-side code in combination with client applications without needing to learn another language. You shouldn't have to rely on all of our users' browsers to upgrade to use the new ECMAScript standards in Node.js because you can choose ECMAScript version use by updating the Node.js edition, and you can additionally facilitate certain experimental features by running Node.js with flags.



```
index.js x ... nodejs-dev-0001-01--3000.local.web...  
1  const http = require('http')  
2  
3  const hostname = '127.0.0.1'  
4  const port = 3000  
5  
6  const server = http.createServer((req, res) => {  
7    res.statusCode = 200  
8    res.setHeader('Content-Type', 'text/plain')  
9    res.end('Hello World\n')  
10 }])  
11  
12 server.listen(port, hostname, () => {  
13   console.log(`Server running at http://${hostname}:$  
    ( ... )`)  
Terminal +  
Server running at http://127.0.0.1:3000/  
[]
```

Fig 12. An example of a Node.js Application

## **Chapter 04: LIVE PROJECT**

### **4.1 Description**

The major task of the project is to integrate social media signup into our application. As we were working on the chrome extension version of the application so my task was to create signup authentication for the extension. In extension, we need to add the server-client connection in the background.js file. This file accepts the API keys and creates a path for the user to connect to the server. After creating the authorization flow for the user, the authorization page asks for the user credentials to log them into our application.

The successful login response returns an access token on the basis of which user information is extracted. We extract the user's name and email from the data collected by parsing the access token. This user information is sent to our server and the user is redirected to the check mail page.

A mail is sent to the user extracted from the tokens. The users can select one of the plans received in the mail. This will give them access to the features selected by the user in their plan.

### **4.2 Method to Create API keys**

We need to create the API keys for different social media for the signup. These API keys will send a request to the server to authenticate the user trying to signup.

#### **4.2.1 Create OAuth Client ID for Google**

1. Go to the Credentials page for the Google API Console.
2. Choose a project, then NEW PROJECT, and give the project a name and, if desired, update the associated project ID. Choose to Create.
3. Choose to Create credential, then OAuth client ID on the Credentials page.
4. If the Consent screen prompts you to enter a product name, click "Configure Consent Screen", enter the necessary information, and click Save to continue to the Credentials screen.
5. For the Application system, choose web application and fill in any further information needed. Choose to Create.



6. Copy the client ID and client secret from the page that displays on your clipboard, since you will need these to configure your client library.

### **4.2.2 Create APP ID for Facebook**

To create a Facebook app, visit [Create Facebook App](#).

Make sure to "Add Product" in your app for Facebook Login. Open settings and specify "https://<extension-id>.chromiumapp.org/" in Valid OAuth Redirect URIs and Allowed Domains for the JavaScript SDK.

To allow the user access to your app, go to App Review->Permission and Features and get advance access to "public\_profile".

### **4.2.3 Create API key and secret for Twitter**

Visit [Twitter Development Portal](#), create a project, and add the app to the project. Save your API key and secret as it will be used later for authentication.

Turn on OAuth 1.0a from User Authentication settings. Add your callback URL in OAuth 1.0a which will be "https://<extension-id>.chromiumapp.org/".

Add the same callback URL in .env.production file:

```
REACT_APP_TWITTER_CALLBACK_URL="https://<extension-id>.chromiumapp.org/"
```

## **4.3 Building and Deploying the Chrome Extension**

This step describes how to build the chrome extension through react app. We also see how to deploy the extension locally as well as through the chrome web store.

### **4.3.1 Build this chrome Extension**

To build the project, proceed with the following steps:

1. Open the project in cmd and run -> npm run build
2. build folder will be generated in the project directory structure, you can deploy this build folder as a chrome extension.

### **4.3.2 Deploy Locally**

1. Go to chrome://extensions/
2. Turn on the Developer mode
3. Click on the "Load Unpacked" button.
4. Place the build folder generated in the building step.

The extension will be loaded in your browser.

### **4.3.3 Deploy on Chrome Web Store**

1. To get started, go to the Chrome Developer Dashboard.
2. Log in to the developer account you selected in Step 1 and click on "Save."
3. Select the button to add a new item.
4. If this is your first time uploading an item, you must first accept the developer contract before proceeding to the next step.
5. Select your zip file and then click Upload. On the next page, you can update your extension if the manifest and ZIP file is valid.

## **4.4 Code**

The basic code of background.js is as follows. I have mentioned the event listener function and created a web authentication flow. I have mentioned the endpoint functions for Google, Facebook, and Twitter used in the background.js.

## 4.4.1 Launch Web Authentication Flow

```
chrome.identity.launchWebAuthFlow({
  url: url,
  interactive: true,
}),
// returns promise only if authorization is successful
function (redirectUrl) {
  // URL will include a parameter that either is an access token or can be exchanged for an access token
  if (redirectUrl) {
    sendResponse({ data: { redirectUrl } });
  } else {
    sendResponse({ err: "Something went wrong." });
  }
}
};
```

## 4.4.2 Google Endpoint Information

```
function getGoogleAuthEndpoint() {
  const nonce = encodeURIComponent(
    Math.random().toString(36).substring(2, 15) +
    Math.random().toString(36).substring(2, 15)
  );

  const googleOpenIdEndpointUrl =
    `https://accounts.google.com/o/oauth2/v2/auth?client_id=${googleLoginInfo.clientId}` +
    `&response_type=${googleLoginInfo.responseType}&redirect_uri=${googleLoginInfo.redirectUrl}` +
    `&state=${googleLoginInfo.state}&scope=${googleLoginInfo.scope} &prompt=${googleLoginInfo.prompt}&nonce=${nonce}`;

  return googleOpenIdEndpointUrl;
}
```

## 4.4.3 Facebook Endpoint Information

```
function getFacebookAuthEndpoint() {
  const nonce = encodeURIComponent(
    Math.random().toString(36).substring(2, 15) +
    Math.random().toString(36).substring(2, 15)
  );

  const facebookOpenIdEndpointUrl =
    `https://www.facebook.com/dialog/oauth?app_id=${facebookLoginInfo.appId}` +
    `&response_type=${facebookLoginInfo.responseType}&redirect_uri=${facebookLoginInfo.redirectUrl}` +
    `&state=${facebookLoginInfo.state}&scope=${facebookLoginInfo.scope} &prompt=${facebookLoginInfo.prompt}&nonce=${nonce}`;

  return facebookOpenIdEndpointUrl;
}
```

#### 4.4.4 Twitter Endpoint Information

```
function getTwitterAuthEndpoint(token) {  
  
  const twitterOpenIdEndpointUrl = `https://twitter.com/oauth/authenticate?oauth_token=${token}`;  
  console.log(twitterOpenIdEndpointUrl);  
  return twitterOpenIdEndpointUrl;  
}
```

#### 4.4.5 Chrome OnMessage Event Listener

```
chrome.runtime.onMessage.addListener((request, sender, sendResponse) => {  
  let identityUrl;  
  if (userSignedIn) {  
    console.log("User is already signed in.");  
  } else {  
    switch (request.message) {  
      case LOGIN_TYPE.GOOGLE:  
        identityUrl = getGoogleAuthEndpoint();  
        break;  
      case LOGIN_TYPE.FACEBOOK:  
        identityUrl = getFacebookAuthEndpoint();  
        break;  
      case LOGIN_TYPE.APPLE:  
        identityUrl = getAppleAuthEndpoint();  
        break;  
      case LOGIN_TYPE.TWITTER:  
        identityUrl = getTwitterAuthEndpoint(request.authToken);  
        break;  
    }  
    showAuthDialog(identityUrl, sendResponse);  
  }  
  return true; // to make the function synchronous  
});
```

4.5 Results

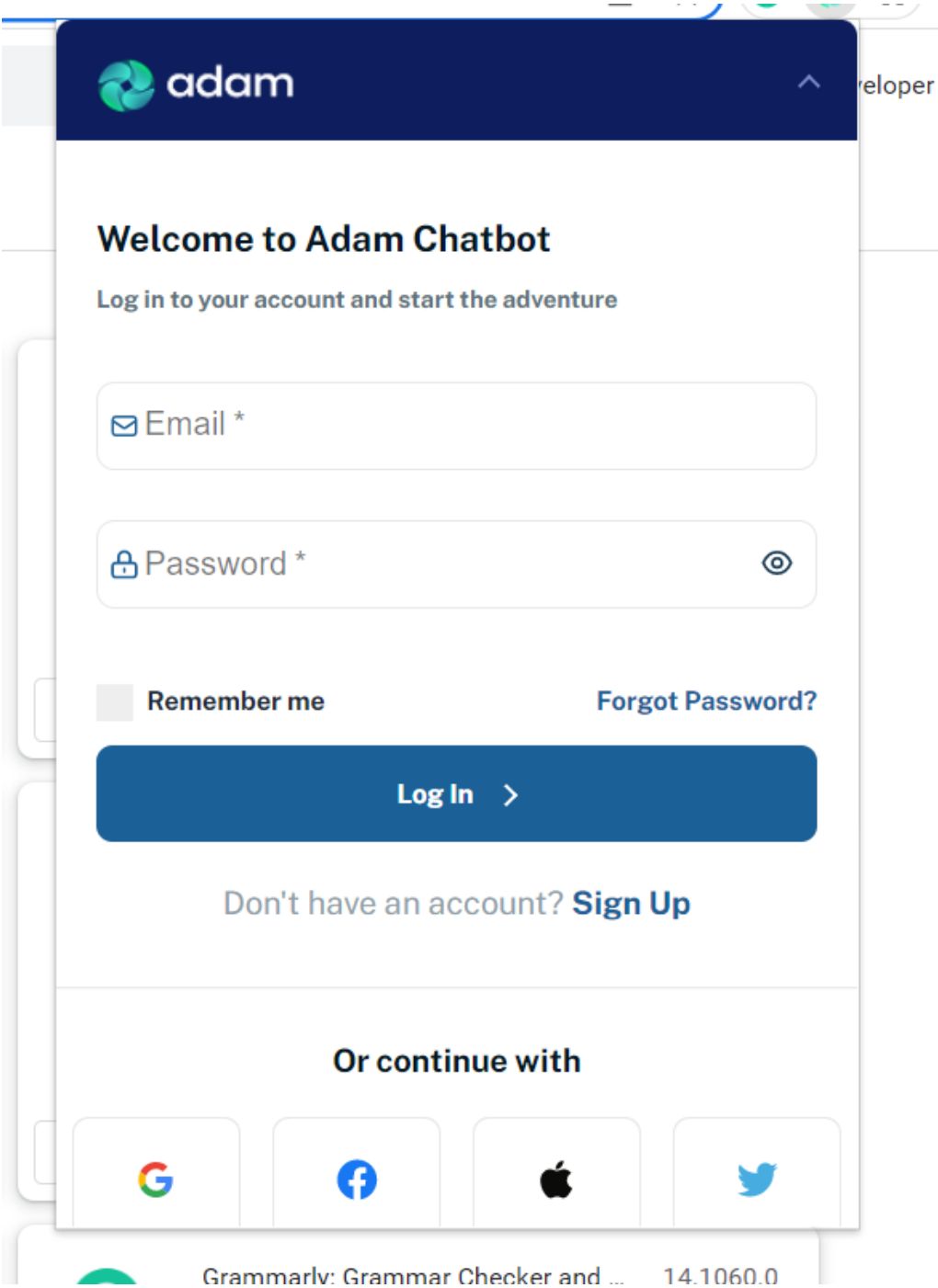


Fig 13. Login Page

## Welcome to Adam Chatbot

Sign up and start the adventure

A white rounded rectangular input field with a light gray border. It contains the text 'Name' with a small envelope icon to its left.A white rounded rectangular input field with a light gray border. It contains the text 'Email' with a small envelope icon to its left.A white rounded rectangular input field with a light gray border. It contains the text 'Password' with a small lock icon to its left and a small eye icon to its right.

Should be at least 8 symbols and contain one small and one big character, special character and number

Remember me

**Sign up >**

Fig 14. Signup Page



Check your mail

We have sent you one-time link to reset password

[Start working >](#)

Did not receive the email?  
Check you spam filter or **try another email address**

Fig 15. Check mail page

## Sunil Garg!

### Activity



79

sentences analyzed and feedback

You were more active than 99% of our users



Sentiment Screen



Account Setting



Widget customization



Go to Adam >

Logout

Fig 16. Home Page



## Chat

User Input

hello

User Intent

hello

Recommendations

1/2 < >

hello and welcome to the club. It is a great club to be a part of. We are here to help each other and have a good time. If you have any questions or want to talk about anything feel free to message me. I am here for you to stay and help you get closer to God and raise your children in the way that you want. God is great and all knowing so He will answer any question you may have. And He is also great in giving life

Sentiment Score

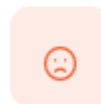
0%



**3%**  
Positive

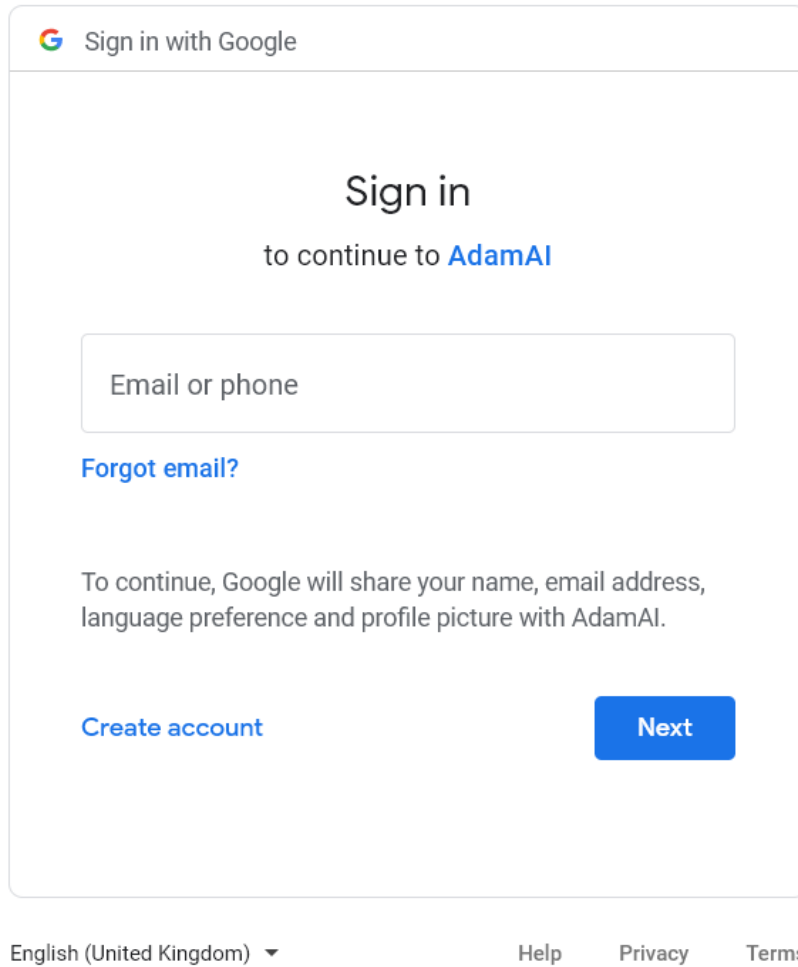


**89%**  
Neutral



**6%**  
Negative

Fig 17. Sentiment Page



Sign in with Google

## Sign in

to continue to [AdamAI](#)

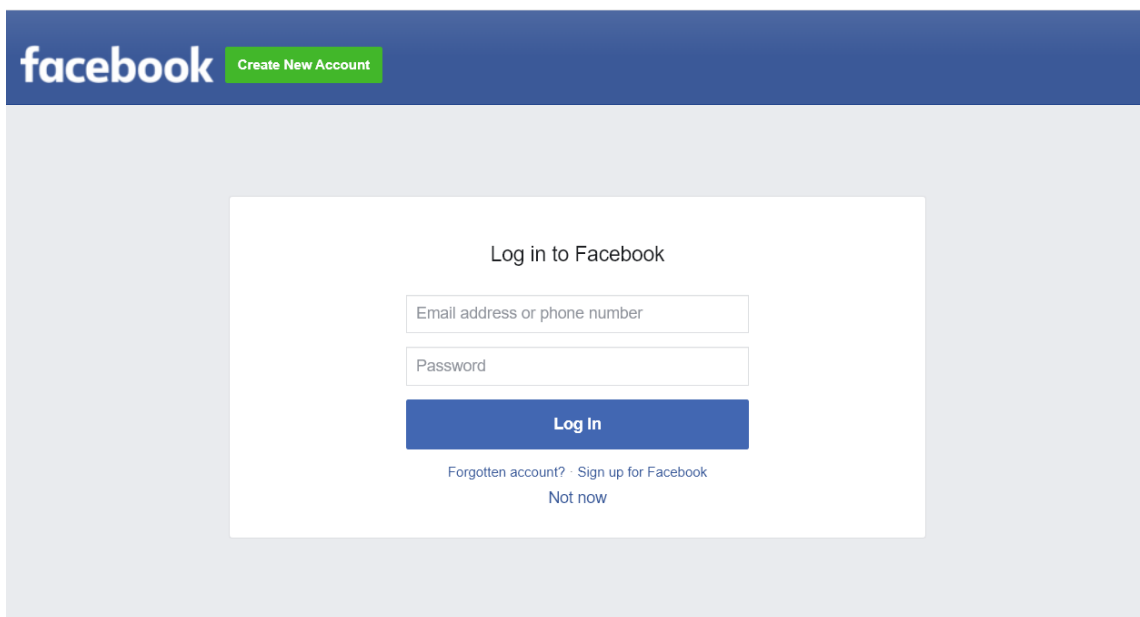
[Forgot email?](#)

To continue, Google will share your name, email address, language preference and profile picture with AdamAI.

[Create account](#) [Next](#)

English (United Kingdom) ▾ [Help](#) [Privacy](#) [Terms](#)

Fig 18. Google Authorization



facebook [Create New Account](#)

## Log in to Facebook

[Log In](#)

[Forgotten account?](#) · [Sign up for Facebook](#)  
[Not now](#)

Fig 19. Facebook Authorization

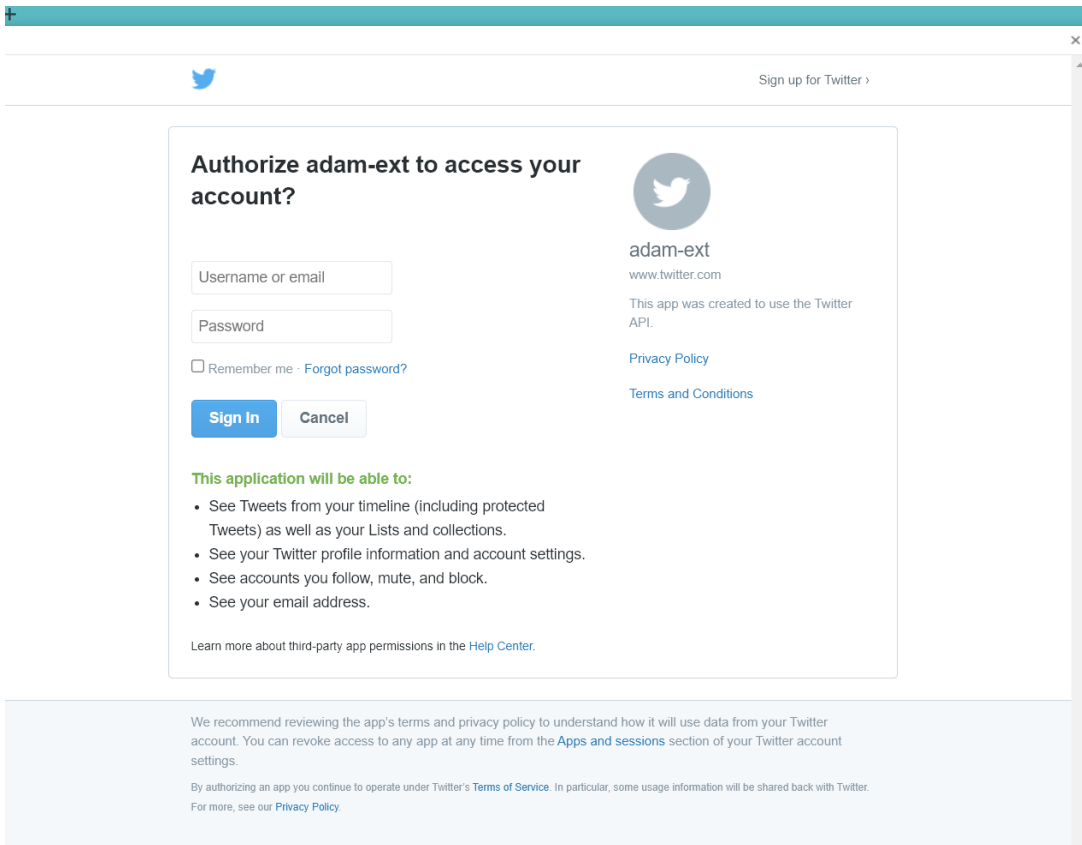


Fig 20. Twitter Authorization

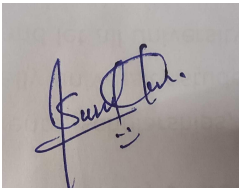
## **Chapter 05: CONCLUSION**

### **5.1 Conclusion**

In conclusion, this internship has become a wonderful and rewarding experience. I can say that my stay with Paxcom India (P) Ltd - A Paymentus Company was quite beneficial to me. Needless to say, the technical components of my work aren't perfect and, given enough time, could be improved. As someone who had no prior experience with MERN stack development, I think the effort I spent learning and understanding about was well spent, as it helped to construct a fully effective web service. Two of the most significant lessons I've learned are time management and self-motivation.

### **5.2 Mentor's Review**

She started training on ReactJS and NodeJS, and within a month she started working on a live project. She is a very talented, quick learner, and open to new skills and approaches.

A handwritten signature in blue ink on a light-colored background. The signature is stylized and appears to read 'Sunil Garg' with a small smiley face below it.

Sunil Garg  
Senior Software Engineer

## REFERENCES

1. <https://www.w3schools.com/html/default.asp>
2. <https://www.w3schools.com/css/default.asp>
3. <https://www.w3schools.com/js/default.asp>
4. <https://www.w3schools.com/nodejs/>
5. <https://www.youtube.com/watch?v=SccSCuHhOw0>
6. <https://www.npmjs.com/package/express>
7. <https://www.mongodb.com/docs/>
8. <https://www.w3schools.com/REACT/DEFAULT.ASP>
9. <https://stackoverflow.com/>
10. <https://github.com/>

# PLAGIARISM REPORT

BHAWNA DEY 181231

## ORIGINALITY REPORT

<b>15%</b> SIMILARITY INDEX	<b>8%</b> INTERNET SOURCES	<b>1%</b> PUBLICATIONS	<b>13%</b> STUDENT PAPERS
--------------------------------	-------------------------------	---------------------------	------------------------------

## PRIMARY SOURCES

<b>1</b>	<b>Submitted to Babes-Bolyai University</b> Student Paper	<b>2%</b>
<b>2</b>	<b>www.coursehero.com</b> Internet Source	<b>2%</b>
<b>3</b>	<b>Submitted to Softwarica College Of IT &amp; E-Commerce</b> Student Paper	<b>2%</b>
<b>4</b>	<b>Submitted to Universidad Anahuac México Sur</b> Student Paper	<b>1%</b>
<b>5</b>	<b>www.javatpoint.com</b> Internet Source	<b>1%</b>
<b>6</b>	<b>en.wikipedia.org</b> Internet Source	<b>1%</b>
<b>7</b>	<b>Submitted to The University of Wolverhampton</b> Student Paper	<b>1%</b>
<b>8</b>	<b>Submitted to Universiti Teknologi MARA</b> Student Paper	<b>1%</b>

9	Submitted to University of Northumbria at Newcastle Student Paper	1 %
10	Submitted to University of Greenwich Student Paper	1 %
11	Submitted to Lovely Professional University Student Paper	1 %
12	Submitted to Management Development Institute Of Singapore Student Paper	1 %
13	Submitted to Baze University Student Paper	1 %
14	Submitted to Coventry University Student Paper	<1 %
15	Submitted to Monash University Student Paper	<1 %
16	Submitted to West University Of Timisoara Student Paper	<1 %
17	semt20.home.blog Internet Source	<1 %

Exclude quotes  On

Exclude matches  < 14 words

Exclude bibliography  On