## **MAJOR-1 PROJECT**

**MID TERM REPORT** on

**“Real Time Log Visualization using DevOps”**

Submitted by:

|  |  |  |
| --- | --- | --- |
| **Name** | **Enrollment No.** | **Branch** |
| Manik Khurana | R110216092 | B.Tech CSE CCVT |
| Babanjot Singh | R110216052 | B.Tech CSE CCVT |
| Ekanshu Dargan | R110216063 | B.Tech CSE CCVT |
| Chhavi Sharma | R110216055 | B.Tech CSE CCVT |

Under the guidance of:

Mr. Harvinder Singh

Assistant Professor

Department of the Virtualization,

School of Computer Science,

University Of Petroleum and Energy Studies.

Dehradun- 248007



**Approved By**

(Mr. Harvinder Singh) (Dr. Deepshikha Bhargava)

**Project Guide Department Head**



School of Computer Science

University of Petroleum & Energy Studies, Dehradun

Synopsis Report (2019-20)

1. **Project Title**

Real time Log Visualization using DevOps.

1. **Abstract**

In this rapidly developing world, an ample amount of pro professionalism is required in every work environment. Therefore, software developers do not have enough time for everything that they are supposed to do. Earlier a simple task used to take weeks to do and now it happens in mere seconds. With this project we aim to present a software solution with the assistance of the latest tools used in agile environments like DevOps and reduce the amount of time and workload for a simple task such as setting up a website. Crucial details of the website will be taken care of by the ELK (Elasticsearch, Logstash, and Kibana) collectively. The website will be up and running using these ELK Logs and with the help of an ansible playbook. This project will be done using only one ansible playbook. This task, if not done this way requires many more such playbooks and a lot more time. The use of docker and docker-compose will be obvious as we are using it to transfer the components on a remote host. That makes this software solution generic and portable. This website in question will be developed using React.js, HTML and CSS.

*Keywords: Ansible Playbook, Elasticsearch, Logstash, Kibana, Git, React.js*

1. **Introduction**

This project serves as an application - a future solution - to a time costly problem of getting logs manually generated with the help of the outdated softwares. Using our solution the developers will be able to save a lot of their precious time and maybe they can then invest it at better places and grow their careers at a greater rate. The Solution is an ansible playbook that performs the functions of setting up docker and docker-compose on the remote host. This ansible playbook takes the docker-compose along with the Nginx, Php and MySql on the remote host. The compose will hence be setup

// change here

This project is an application - a future product - that can help any student build and download watermark-less resumes and cover letters. This project keeps in mind the basic mindset of students and how well they will adapt to the new software. Using resuArchitect is effortless and secure and its volatility lies in its functionality and its generic character. The resuArchitect is a generic product in its character as the student needs to input her/his details only once and then these details will be fitted in the desired template as required by the student. The resume can then be easily downloaded from the same page and modified as per the student’s need. The Project resuArchitect relies on Node.js’ Single Thread, non-blocking I/O Technology that uses Event loop to process requests. The frontend functionality is controlled and moderated by React.js and the basics of website portal creation - HTML and CSS. The version releases and the entire development is based on agile methodologies, taken care using the famous DevOps Tool - Jenkins. The development will hence follow Continuous Integration/Continuous Deployment (CI/CD) model. We are planning to host the entire development cycle on Amazon Web Services (AWS)

* **Backend** - Node.js and NPM
* **Frontend** - HTML5, CSS, React.js
* **DevOps Tool -** Jenkins

The Project uses Semantic Versioning System (Semver) for naming the new releases of resuArchitect.

1. **Literature Review**

Dunlu PENG†, Lidong CAO, Wenjie XU School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China [1] examined that JSON is a light-weight key-esteem style information ex-evolving group. As we probably are aware, the effectiveness of mapping information between various information models is the key point to improve the presentation of web administration applications.

Naimul Islam Naim IMetropolia University of Applied Sciences,30 May 2017 [2] explored that ReactJS as a good stage to be embraced where there are a few choices to pick from. The basics, center engineering, highlights, information dealing with strategies, prominence,what's more, adoptability were examined in this examination. Despite the fact that there is no limitation to utilize ReactJS over different structures it is prescribed to utilize it as a developing web innovation to be embraced by relying upon the idea of the proposed application to be manufactured.

Kenneth Lewenhagen, Anders Åkesson .“Node.js in Open Source projects on Github.” *Blekinge Institute of Technology* [3] performed a study with an aim to provide an insight into how Node.js is used and the Node.js technology adaptation in the open source community. This research displays the diversity of Node.js.

Arpitha R & Mrs. Kavitha S N[4] tested the exhibition of the gadgets or item requires significant investment and needs human help. This can be explained utilizing Continuous Integration (CI) method, it is the most generally utilized procedure by the engineers to incorporate their code into vault, CI is the most proficient and quicker approach to computerize the things. Jenkins is a most effective device which gives constant coordination.

Jurgen Cito, Vincenzo Ferme, and Harald C. Gall, *Dept. of ISE, R. V. College of Engineering, Bengaluru, India.*[5] presents how Docker containers can overcome these issues and aid the reproducibility of research artefacts in software engineering and discusses their applications in the field.

Darshita Kalyani, Dr. Devarshi Mehta [6] examined that Elasticsearch is utilized for use cases like analytics store, auto completer, spell checker, cautioning motor, also, as a universally useful archive store; Full content pursuit is one of it. It is a hearty web index that gives a snappy full content inquiry over different records. It look inside full content fields to discover the report and return the most pertinent outcome first. The pertinence of reports is great as Elasticsearch employments boolean model to discover archive

In his article, “Open-source Scholarship”, Kris Shaffer[7] argues that the open-source software model has lessons to offer the academic community. Here, Kris demonstrates how a scholar can put open-source philosophy into practice using a specific tool developed by and for the community: GitHub.

1. **Problem Statement**

Majority of the resumes are imperfect and they lack a certain appeal that the hirer of any recruiting company looks in a CV or a resume. This is a major reason for unemployment amongst highly skilled engineers.

1. **Objective**

The primary objective of building resuArchitect is to empower students to make appealing and attractive resumes and cover letters in the least time possible. Thereby giving them more time to focus on non-trivial matters.

1. **Methodology**

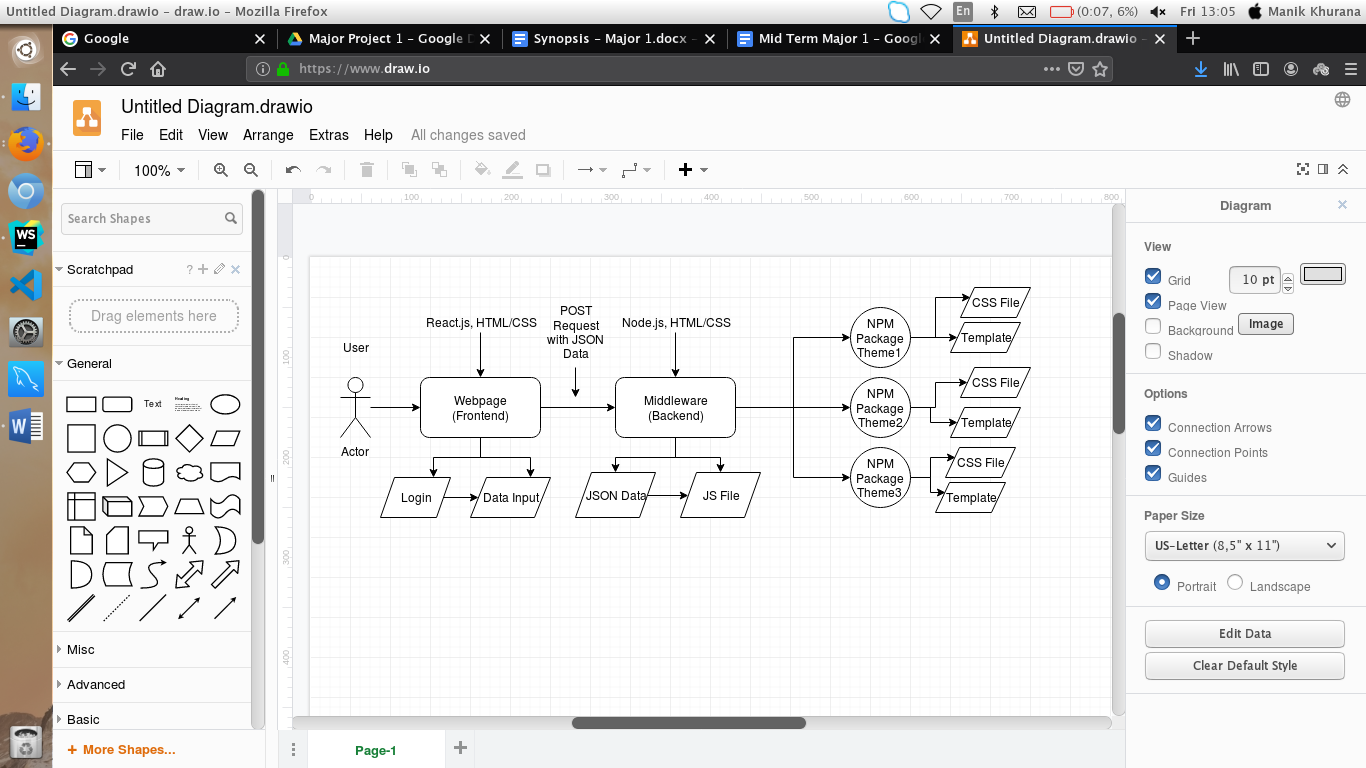
****

Figure 7.1 Flow Chart of the Entire System

**Module 1: Setting up server side application code using Node.js - Express, Sequelize & NPM**

Firstly, the server setup will be done using Node.js - Express. This server will enable us to take in all the API Calls from the frontend. We will here encode and expose different functionalities as callback functions to GET/POST Calls via various routes setup. Every time an API is hit, the user will be getting an appropriate response and his/her resume file with all the details filled.

Whenever the user clicks on the download button for any particular resume format, the server will receive the user’s details in JSON format in a POST Request. The details will then be sent to the callback function in the request itself and this callback function will then take in all the user details and insert it into the desired template. After insertion is done the resume will be sent to the user after automated testing (at Node end via mocha) via sendFile method.

The backend will also use Sequelize ORM to map the user details into the desired database.For now the dialect will be set to MongoDB as it is really efficient to store JSON Data for it is a NoSQL DB. The database will be used to fetch the User Authentication details and their other details in JSON format as stored.

Here is a snapshot of the basic server setup code and a browser snapshot that shows the output.

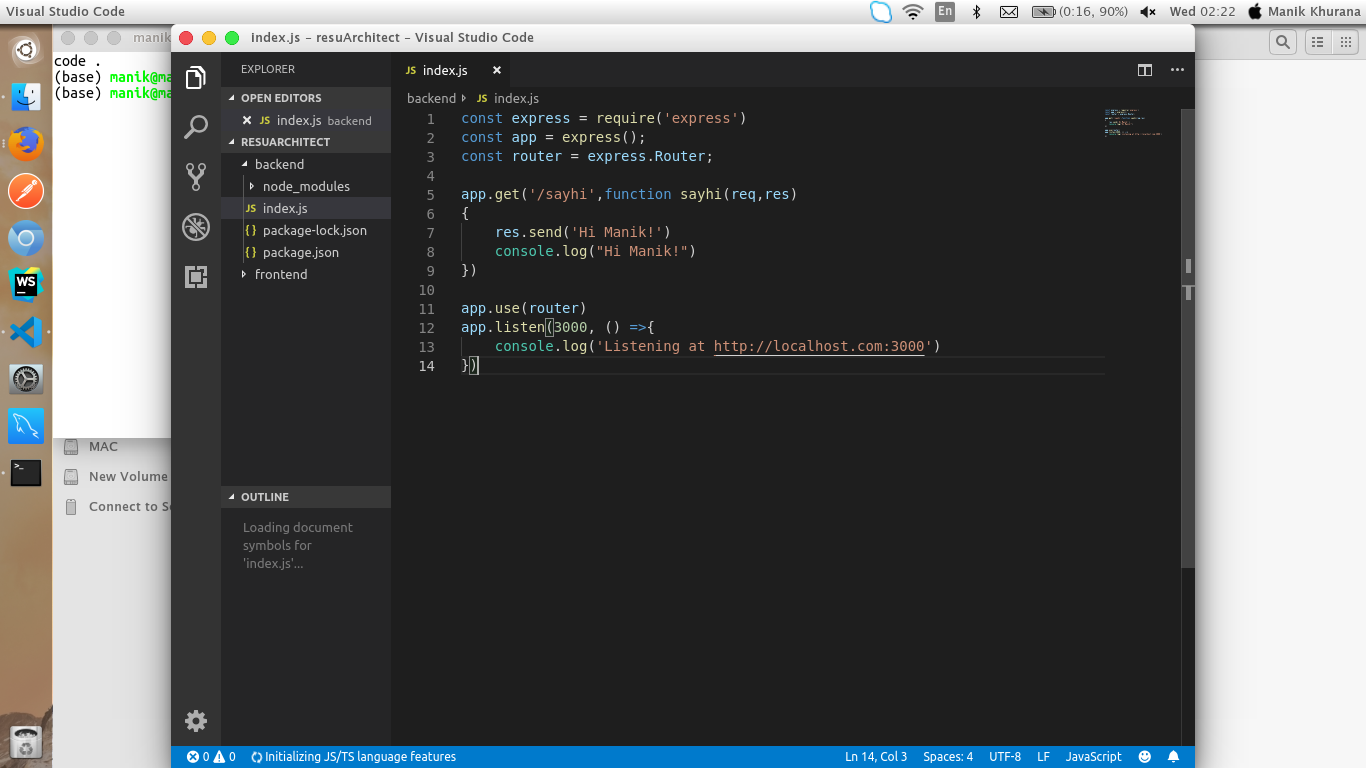
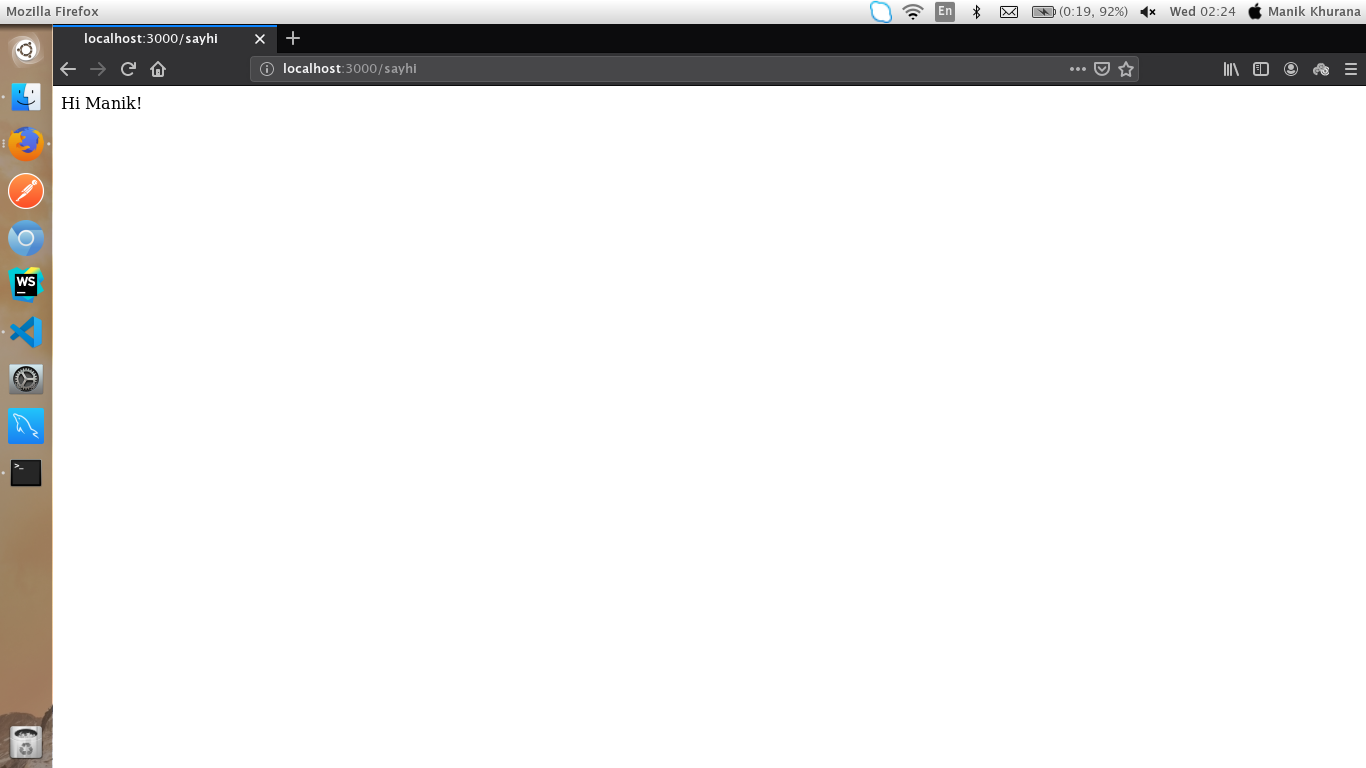
 

Figure 7.1.1 Basic Express Server code snapshot Figure 7.1.2 Logging into the PORT

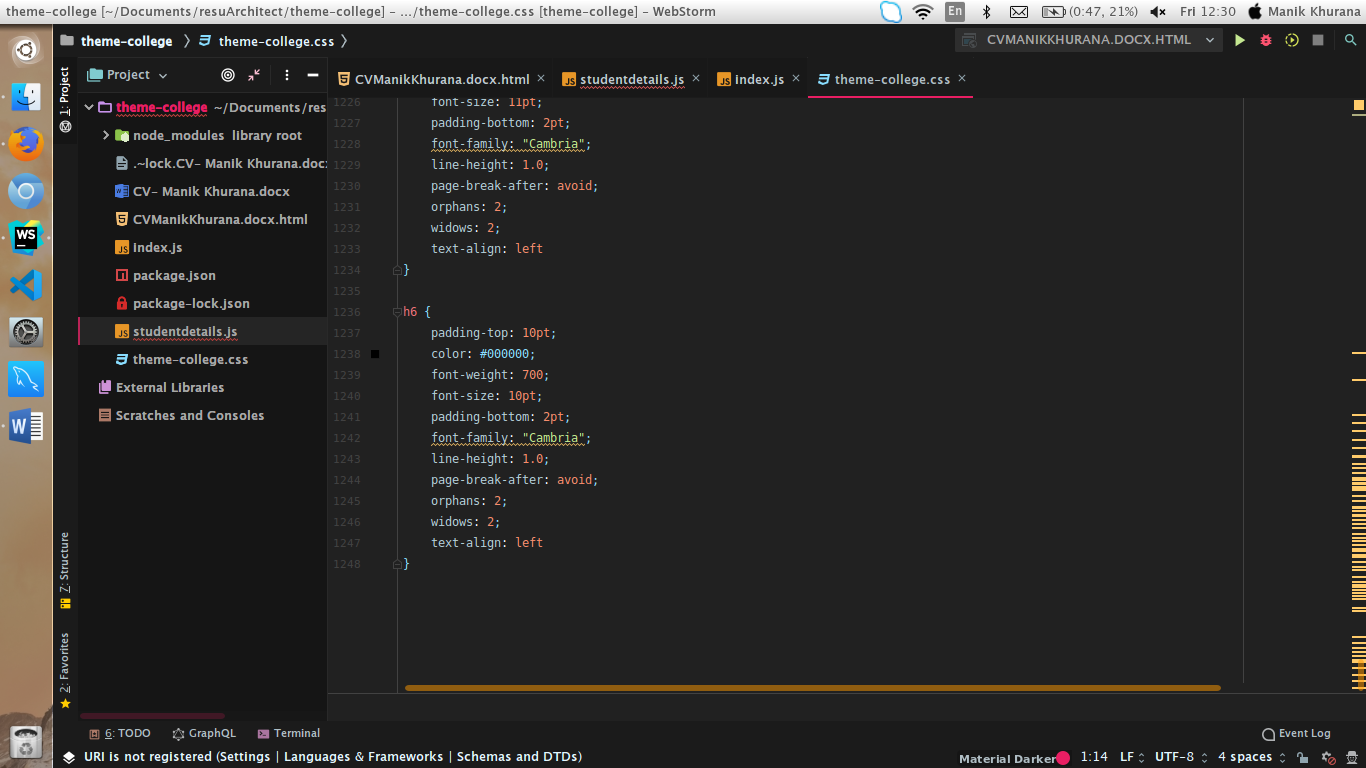


Figure 7.1.3 The theme-college.css file

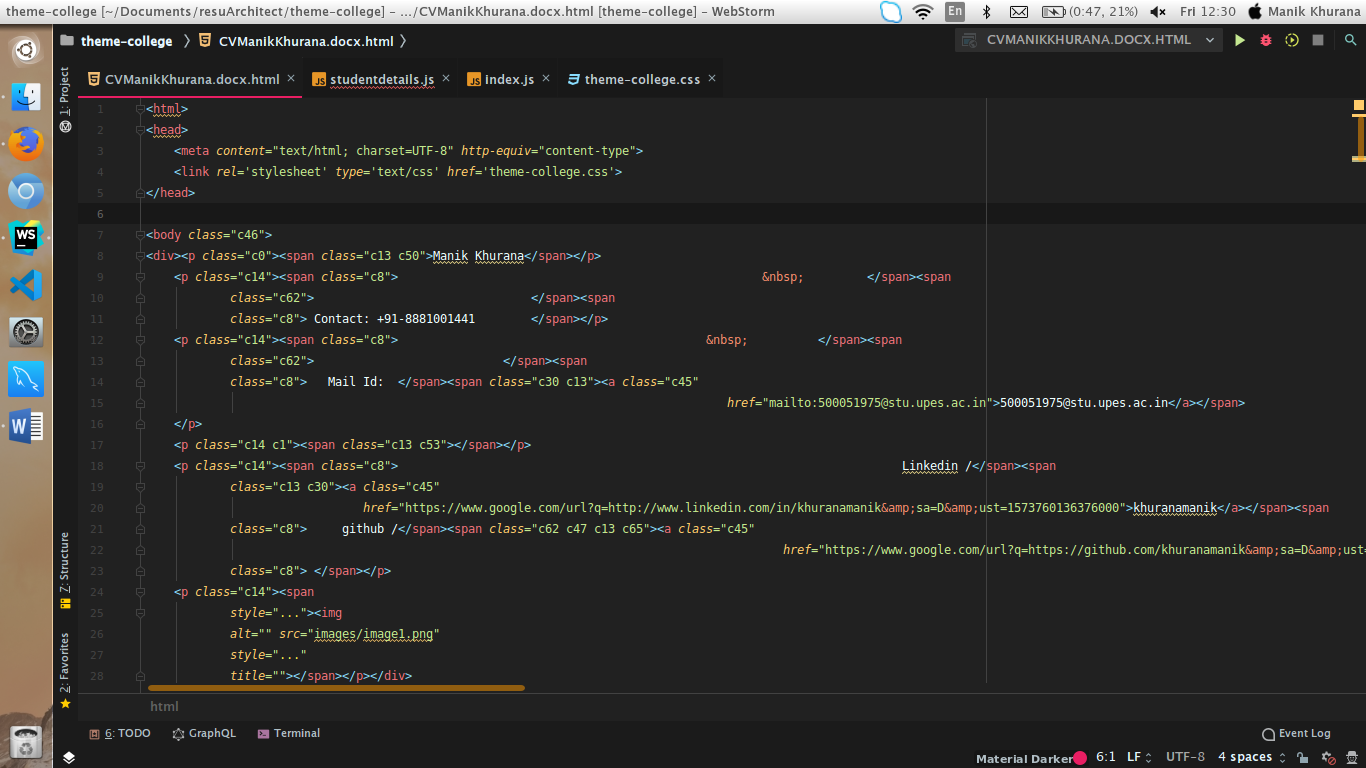


Figure 7.1.4 The HTML File of the CV

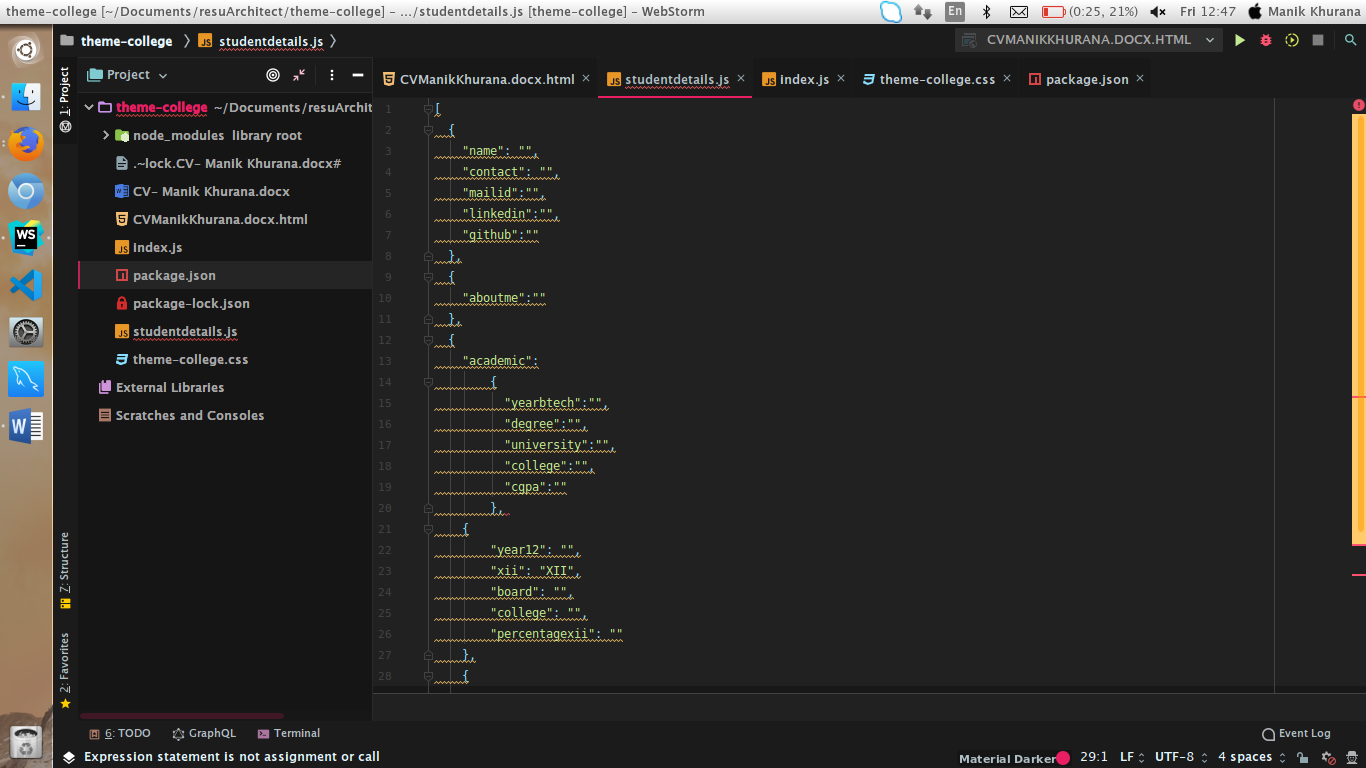


Figure 7.1.5 The JSON File used for communication

**Module 2: Setting up client side application code using React.js, Redux and HTML/CSS**

This is where we build the frontend part of the project. This is where the user will be required to Authenticate himself/herself via various options of Logging in via Google, Linkedin and simply via Email and Password. The user will then be asked various details about his/her Education, Work Experience, etc. These are the details that go into the resume straightaway. In the Downloads option the user will be able to see various themes. The user can then download the resume of his/her choice . When the user will press the download button for a particular theme, his details will be sent in JSON format to the backend API Gateway URL specified as the endpoint for the particular theme.

This way the users can enter their details only once and download any resume format, any number of times. User details will be stored in Database

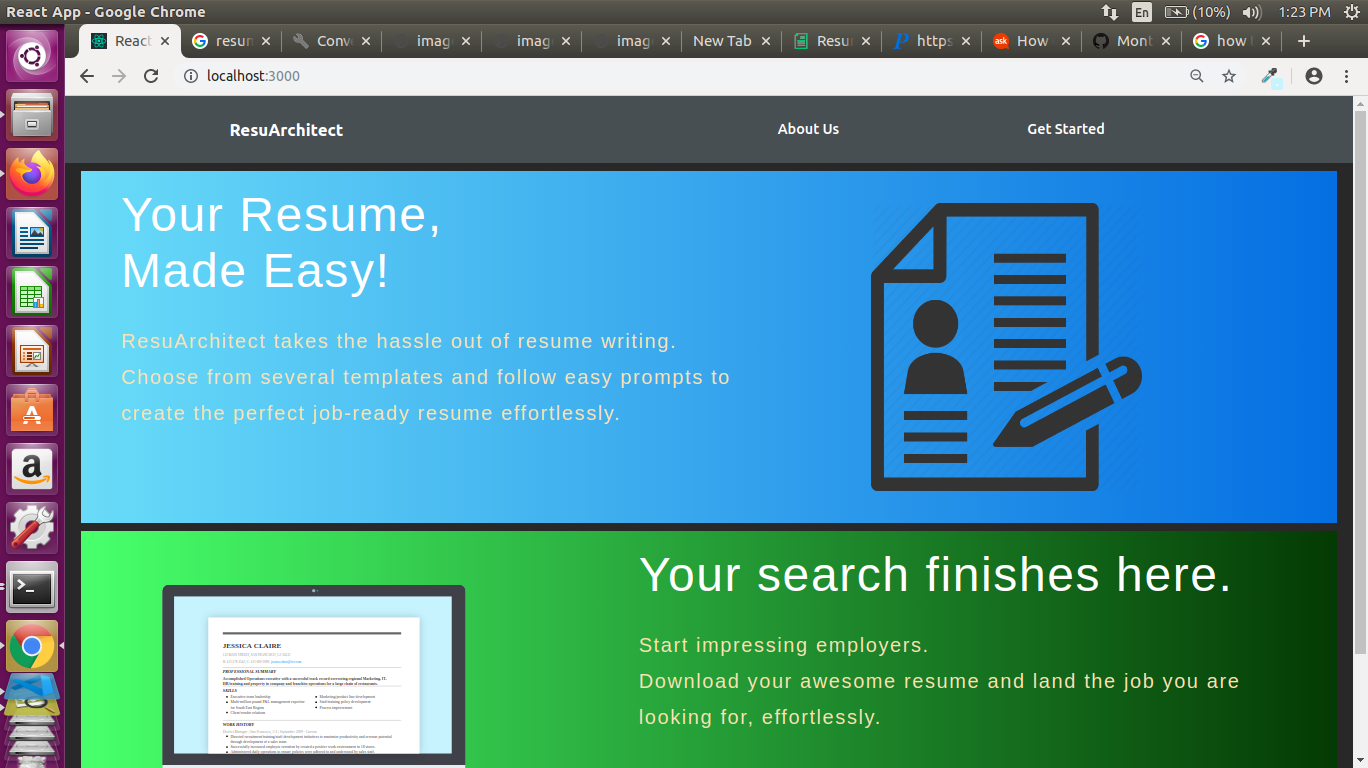


Figure 7.2.1 Main Homepage

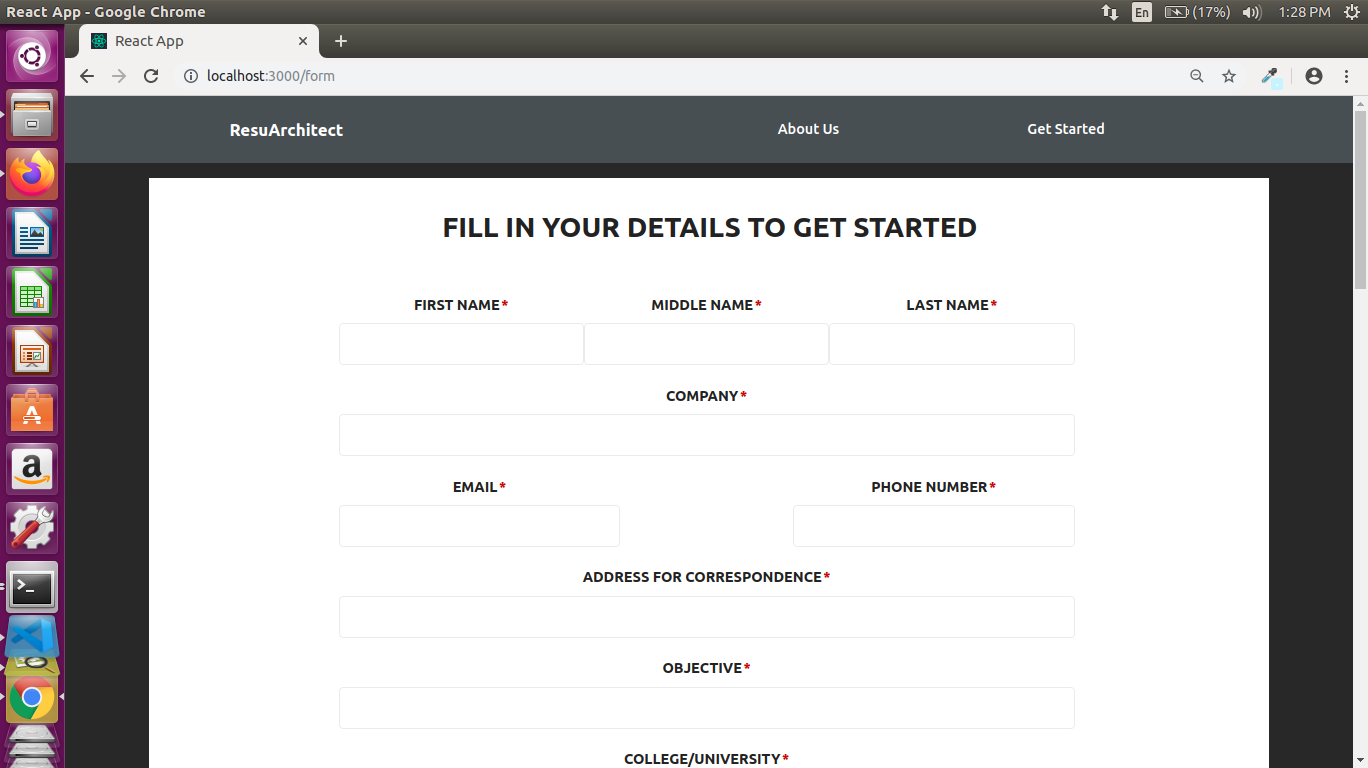


Figure 7.2.2 Form to input data

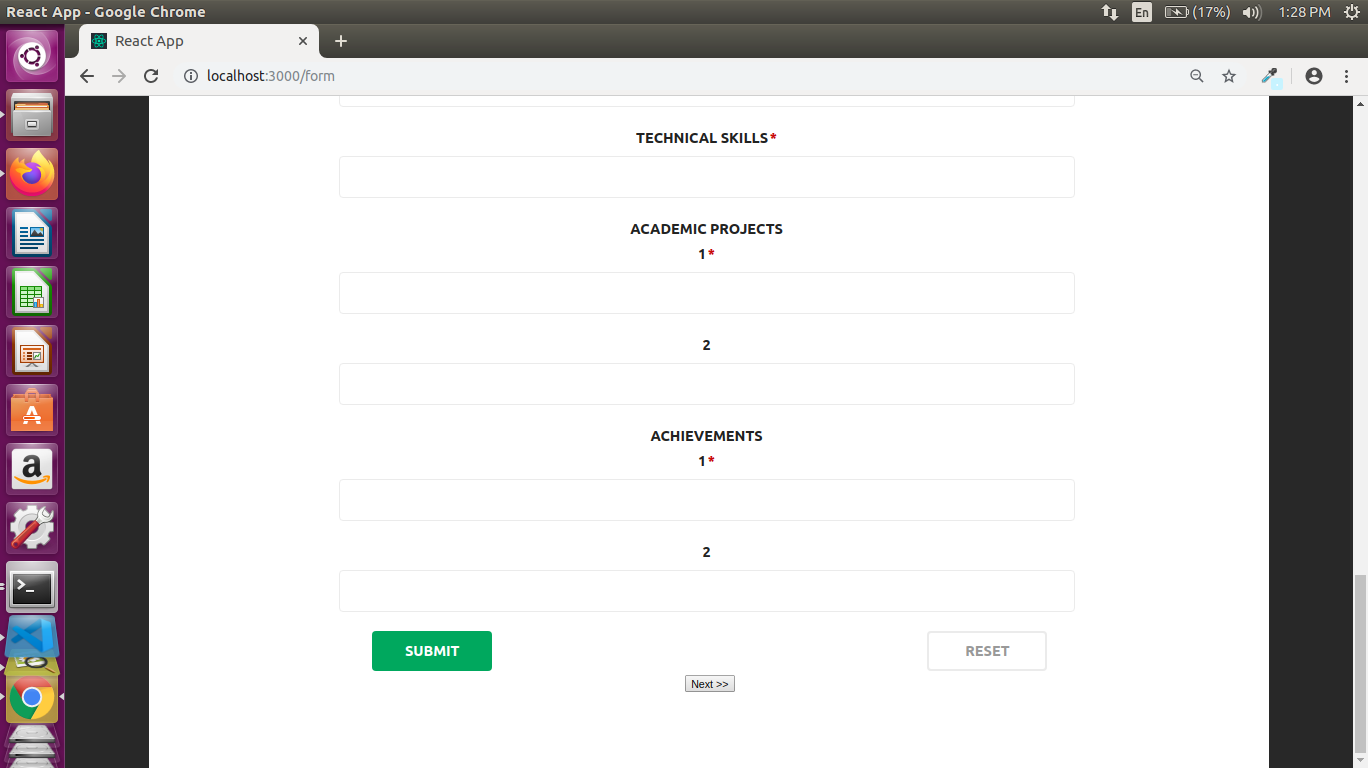


Figure 7.2.3 Submit and Reset Features

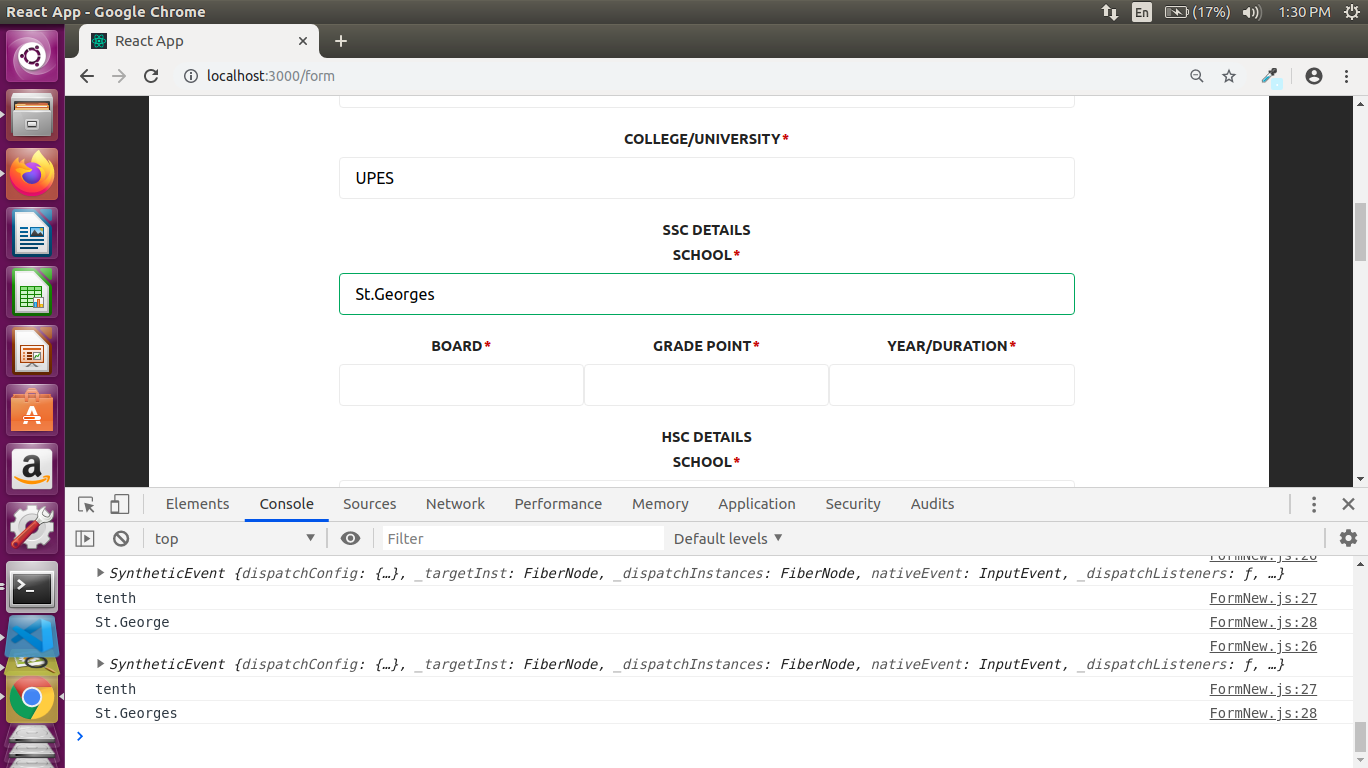


Figure 7.2.4 Console View

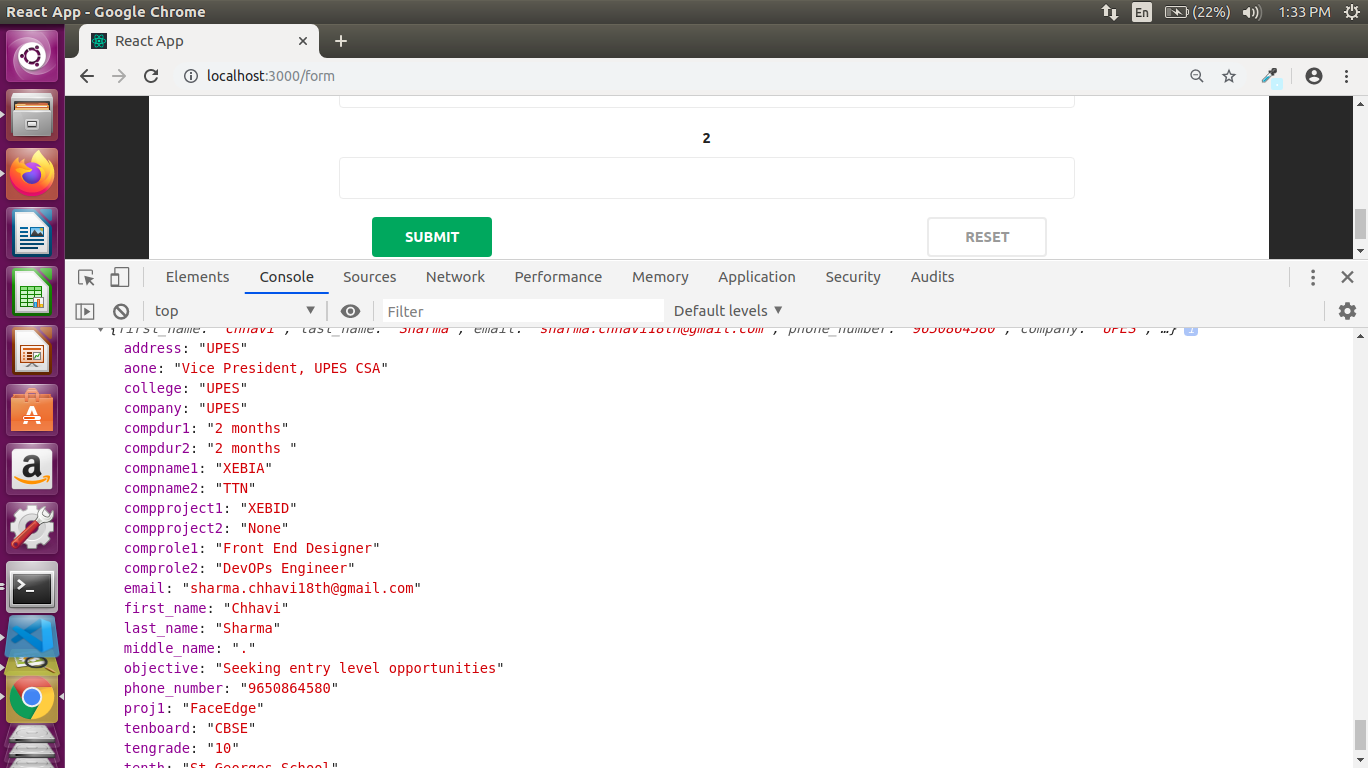


Figure 7.2.5 JSON Object Creation

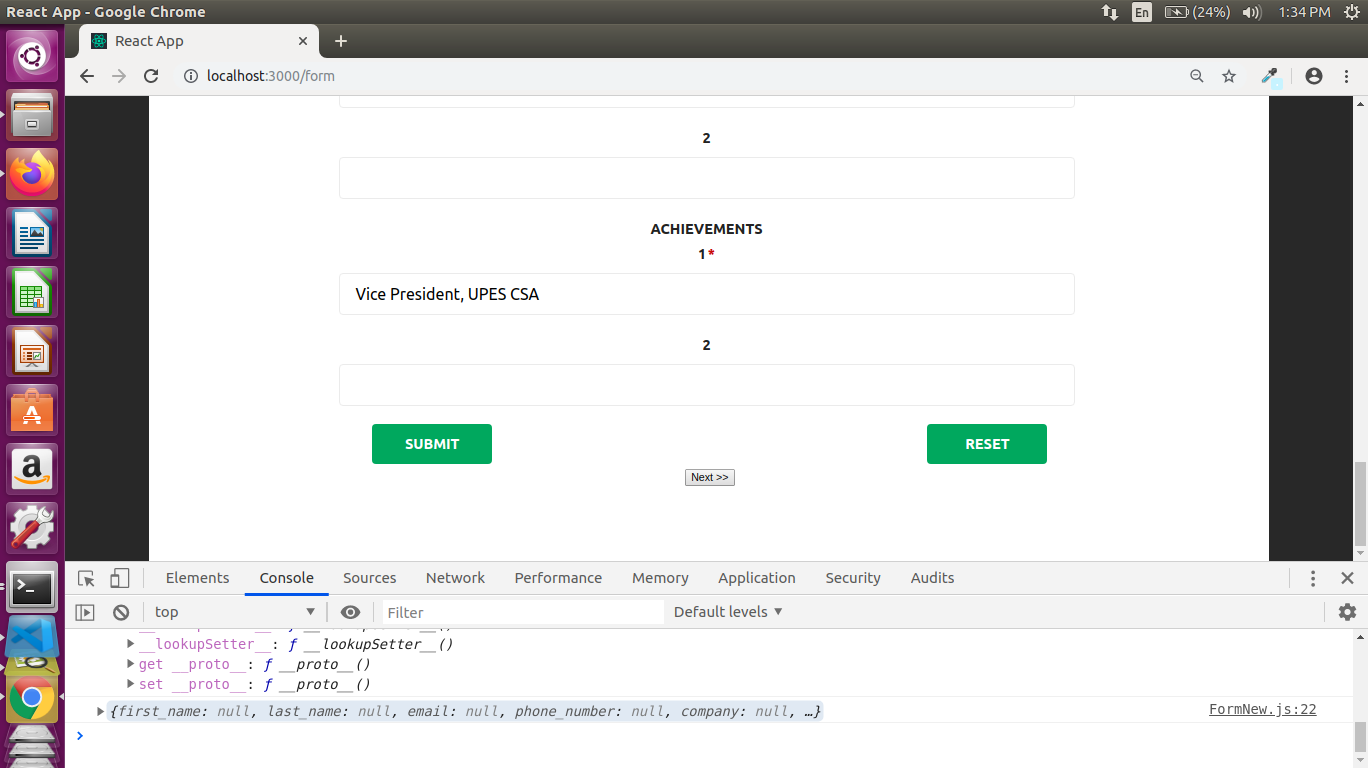


Figure 7.2.6 Reset Feature

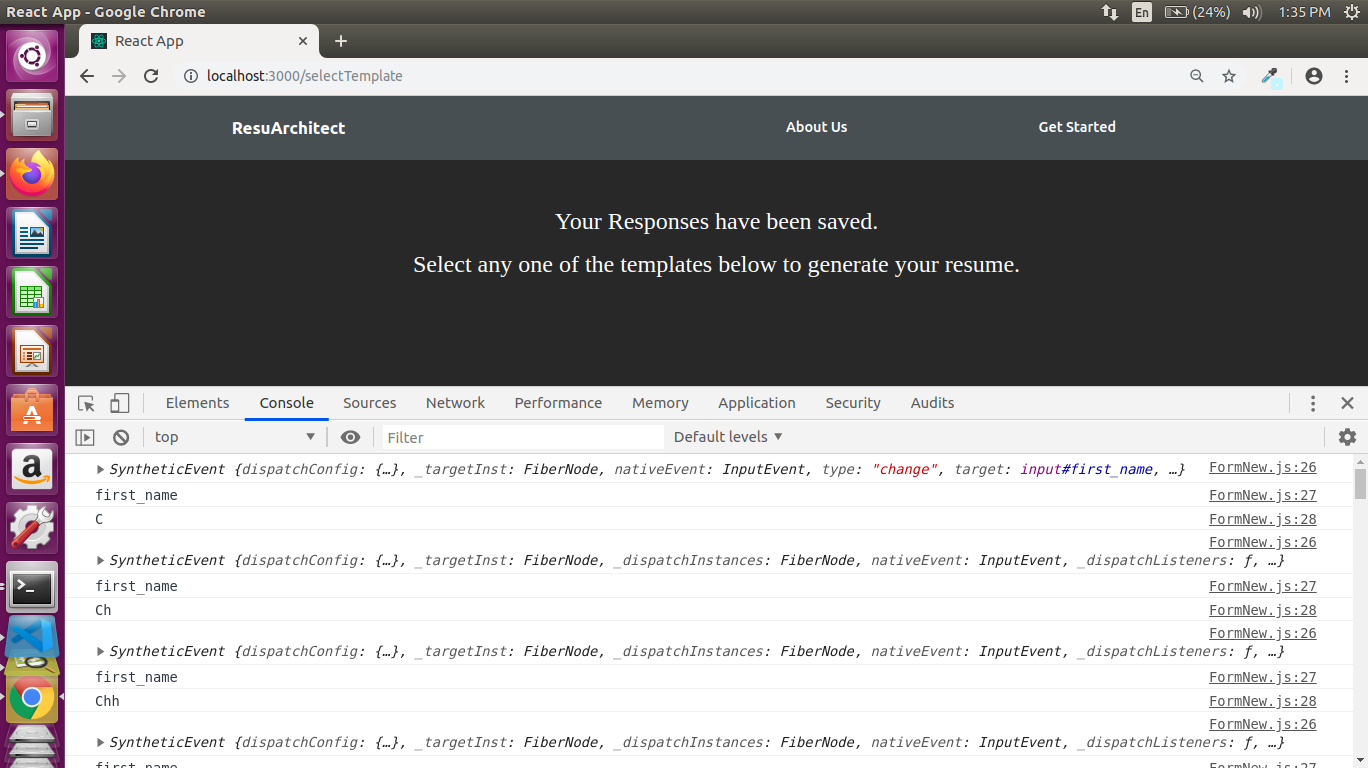


Figure 7.2.7 Responses

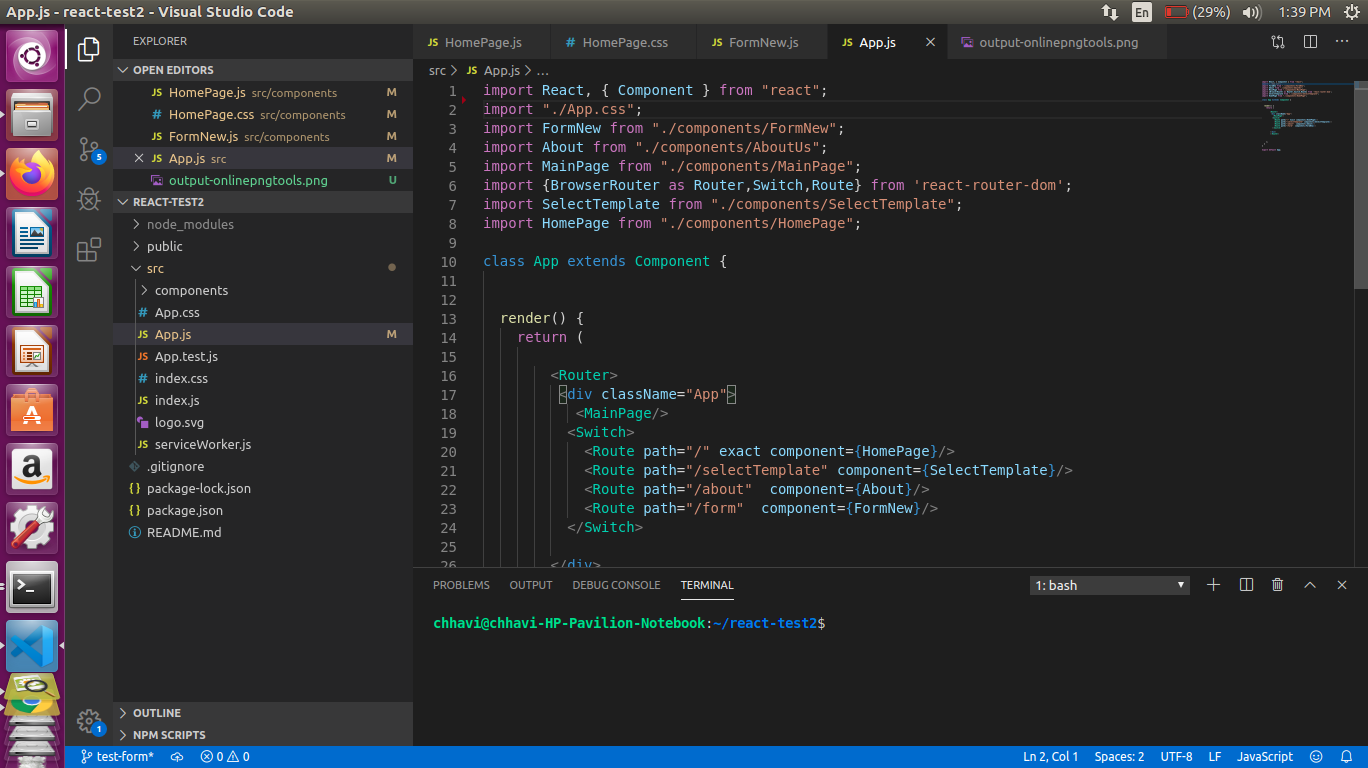


Figure 7.2.8 App.js

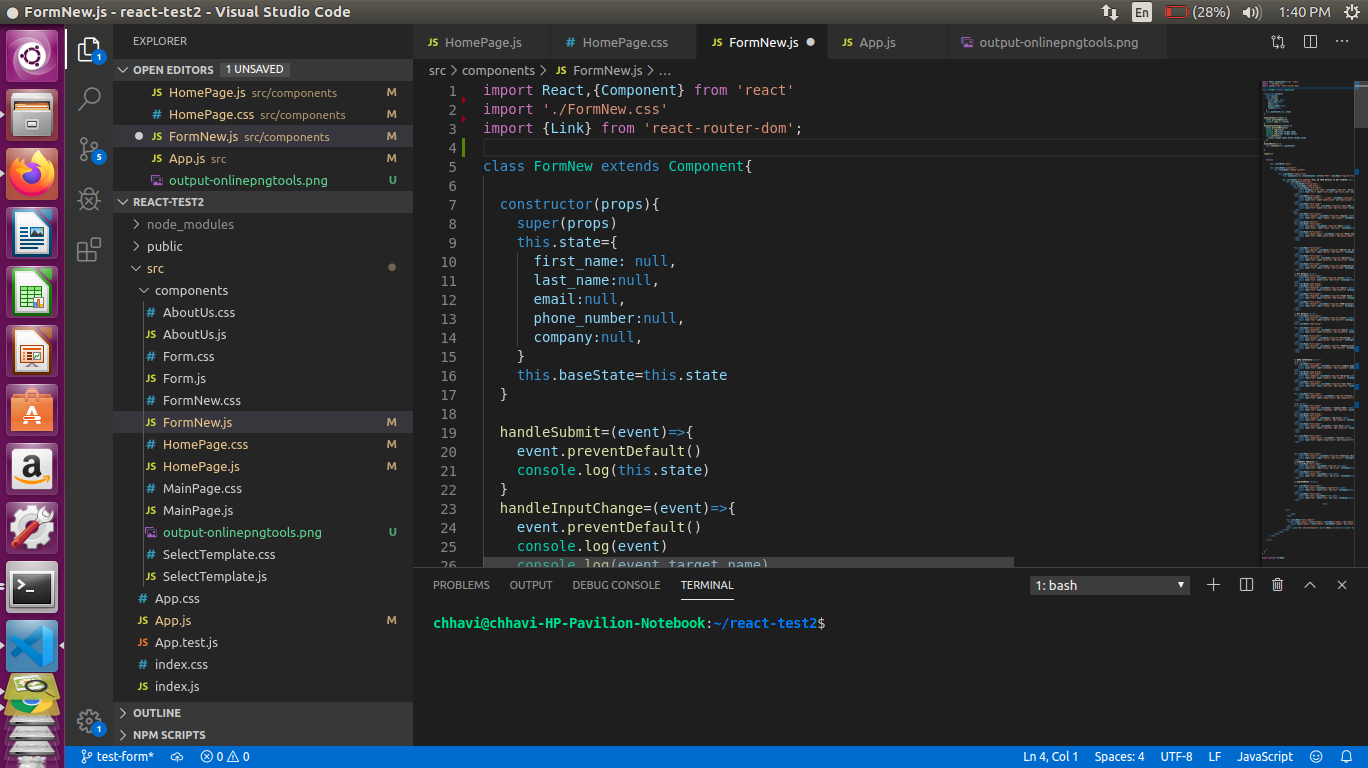


Figure 7.2.9 Form.js file

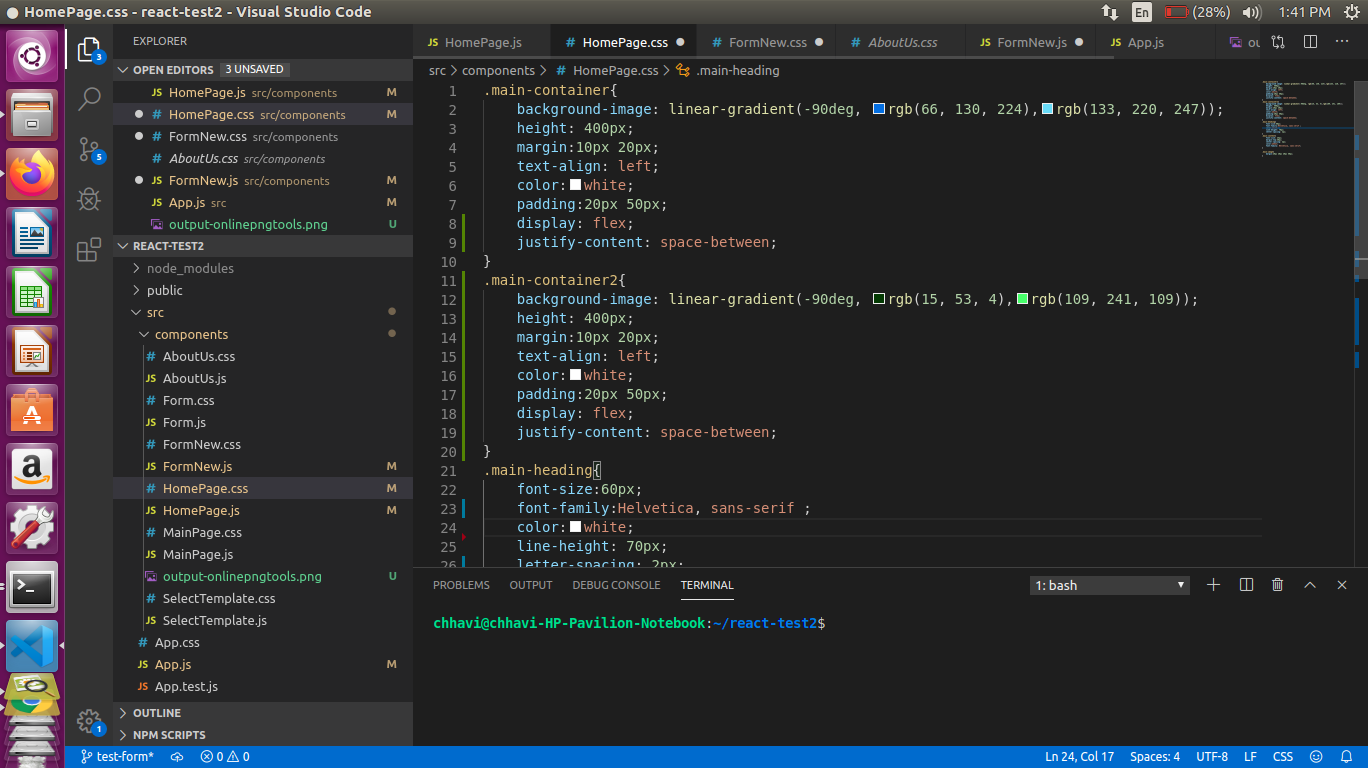


Figure 7.2.10 CSS File

**Module 3: Automation of release and versioning control using Jenkins and Github**

In this module we will implement the building of our web application using Jenkins in Docker.  
We will pull the Node.js and React.js codes from our GitHub repository which will be accompanied by a test to check that the application renders satisfactorily.We will define the ports on which our web application and jenkins will run. Following steps will be followed :-

1. **Run Jenkins on Docker:** We will run Jenkins as a Docker container from the jenkinsci/blueocean Docker image.
2. **Accessing the Jenkins/Blue Ocean Docker container:** Here we will access Jenkins and start the wizard on local host to start it up and do further build operations. We will add the plugins that will be required and suggested and will create the first administrator user. After this we will save and finish.
3. **Fork and clone our repository on GitHub:** In this we will clone our GitHub repository by signing in github and then we will pull the source code of our web application.
4. **Create Pipeline project in Jenkins:** Now we will create our Pipeline that will automate building your Node.js and React application in Jenkins. This Pipeline will be created as a Jenkinsfile, which will be committed to locally cloned Git repository and also it will create a docker image for our Node.js and React.js
5. **Defining Stages in Jenkins:** Here will create all the stages for our pipeline such as Build,Test and Deliver in which all the steps will be covered for the continuous integration to delivery.

**Steps to Create Initial Pipeline as JenkinsFile**

Step 1: A docker file to pull the jenkins container from dockerhub repository

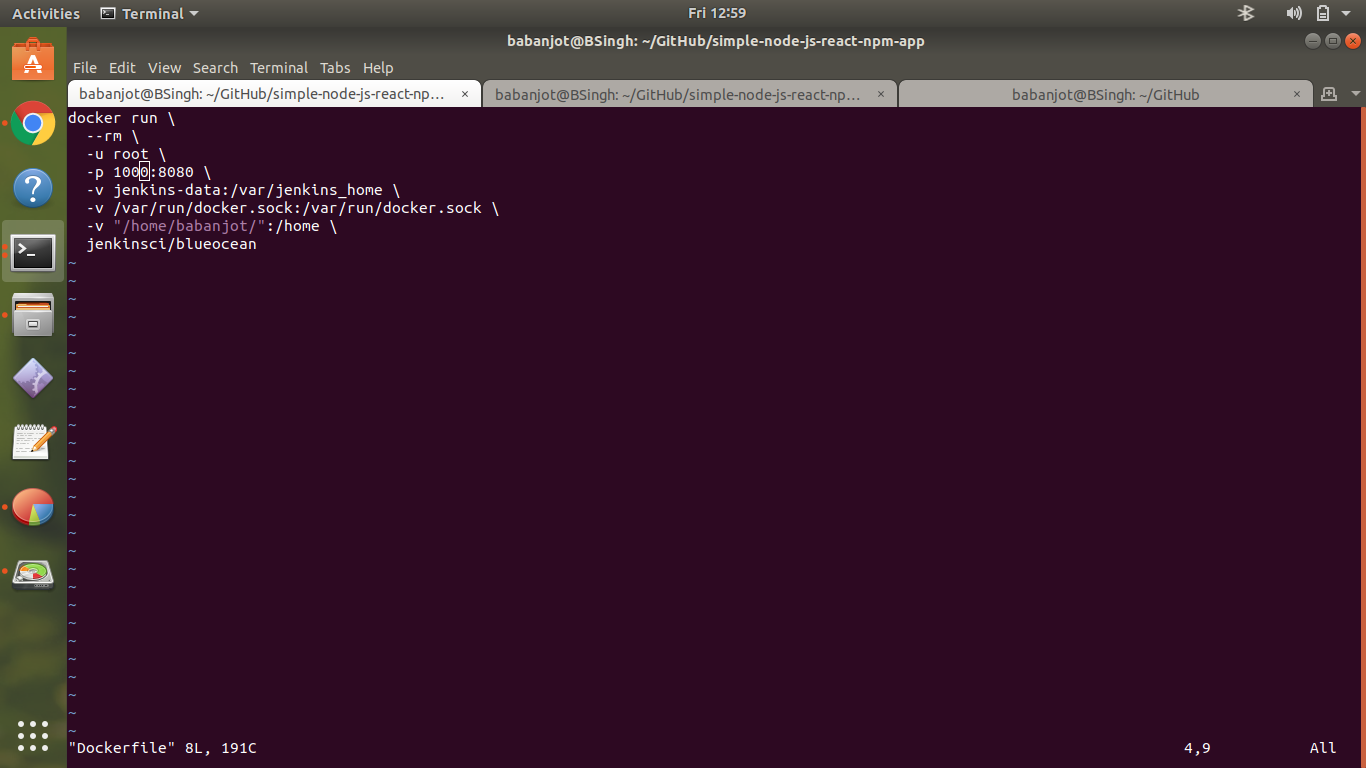


Figure 7.3.1 Docker File

Step 2: Running Docker container and setting up Jenkins on a local host machine

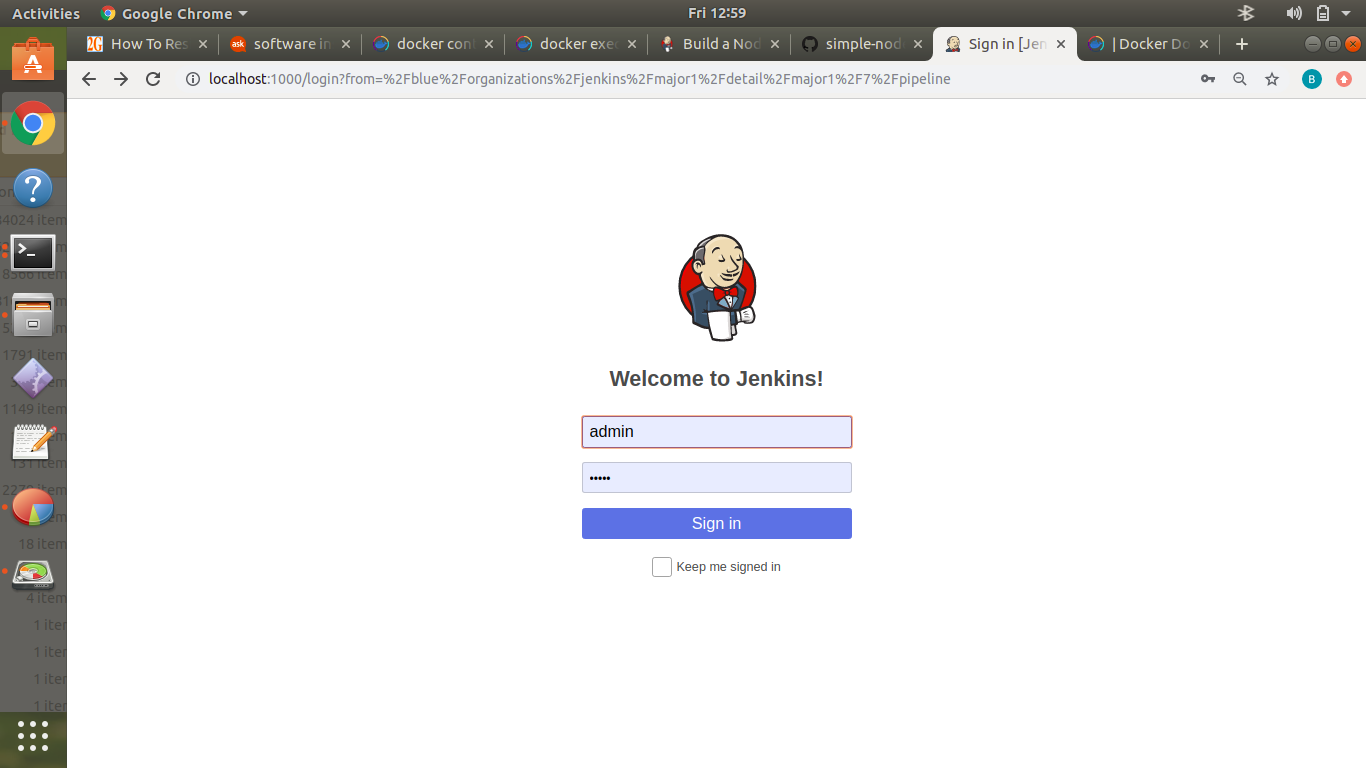


Figure 7.3.2 Welcome to Jenkins

Step 3: Fork and then Clone the sample repository on GitHub Account

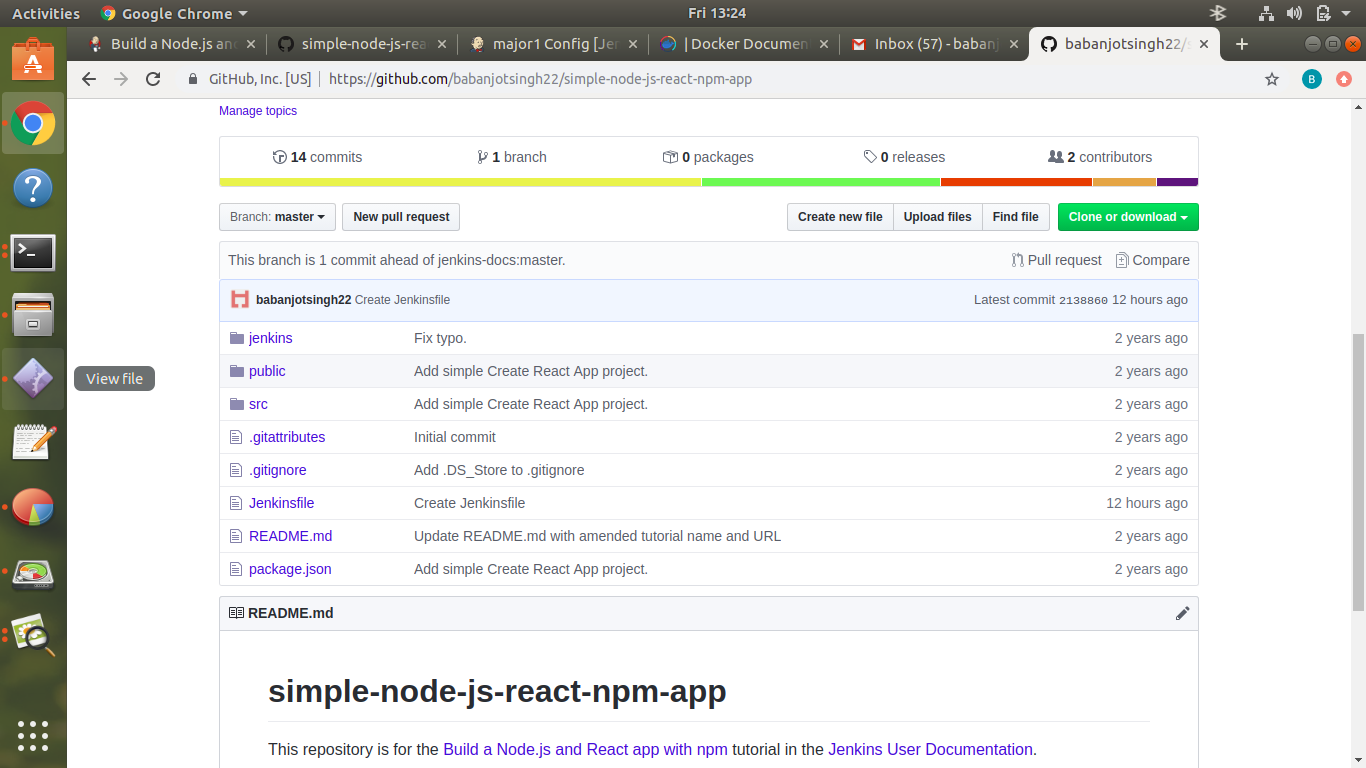


Figure 7.3.3 Cloning GitHub

Step 4: An Entry level Pipeline demonstrating How to use Jenkins to build a simple node.js and react.js application with NPM

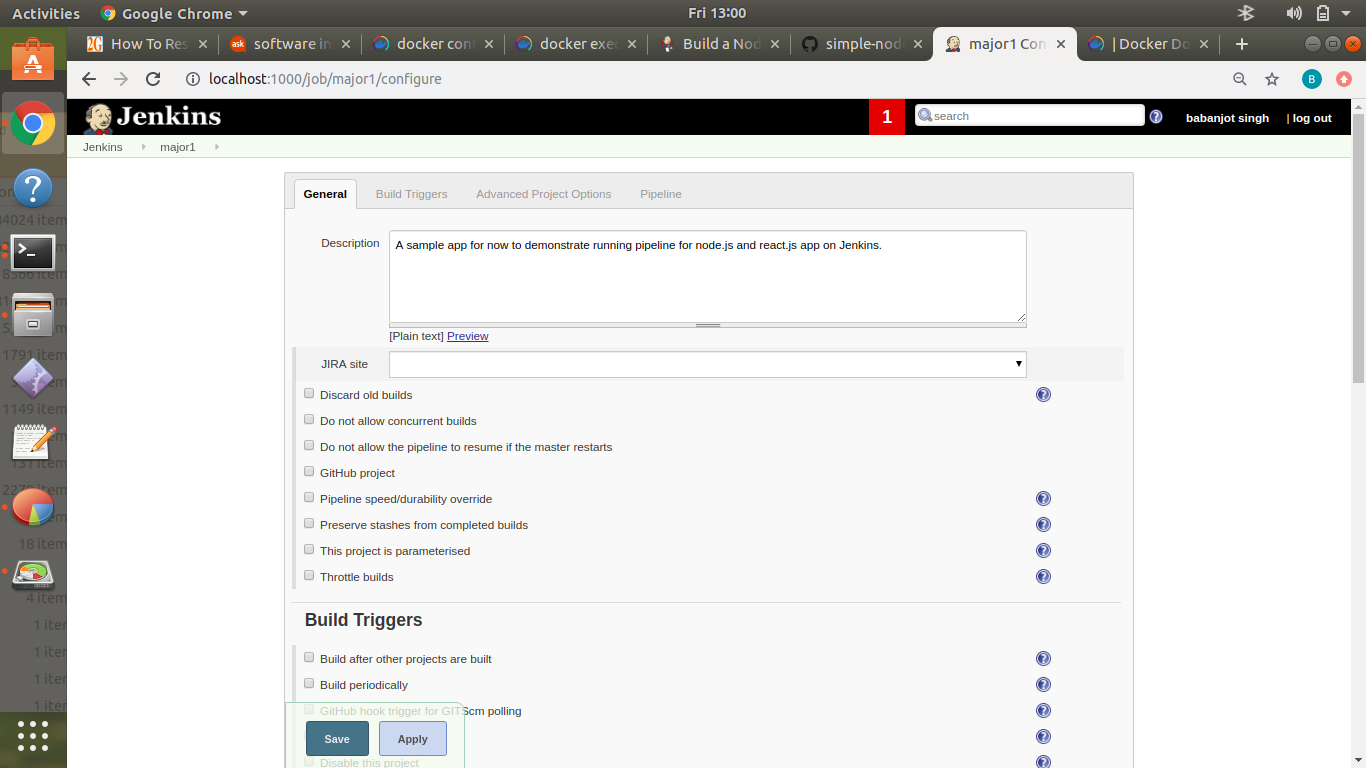


Figure 7.3.4.1 Pipeline in Jenkins

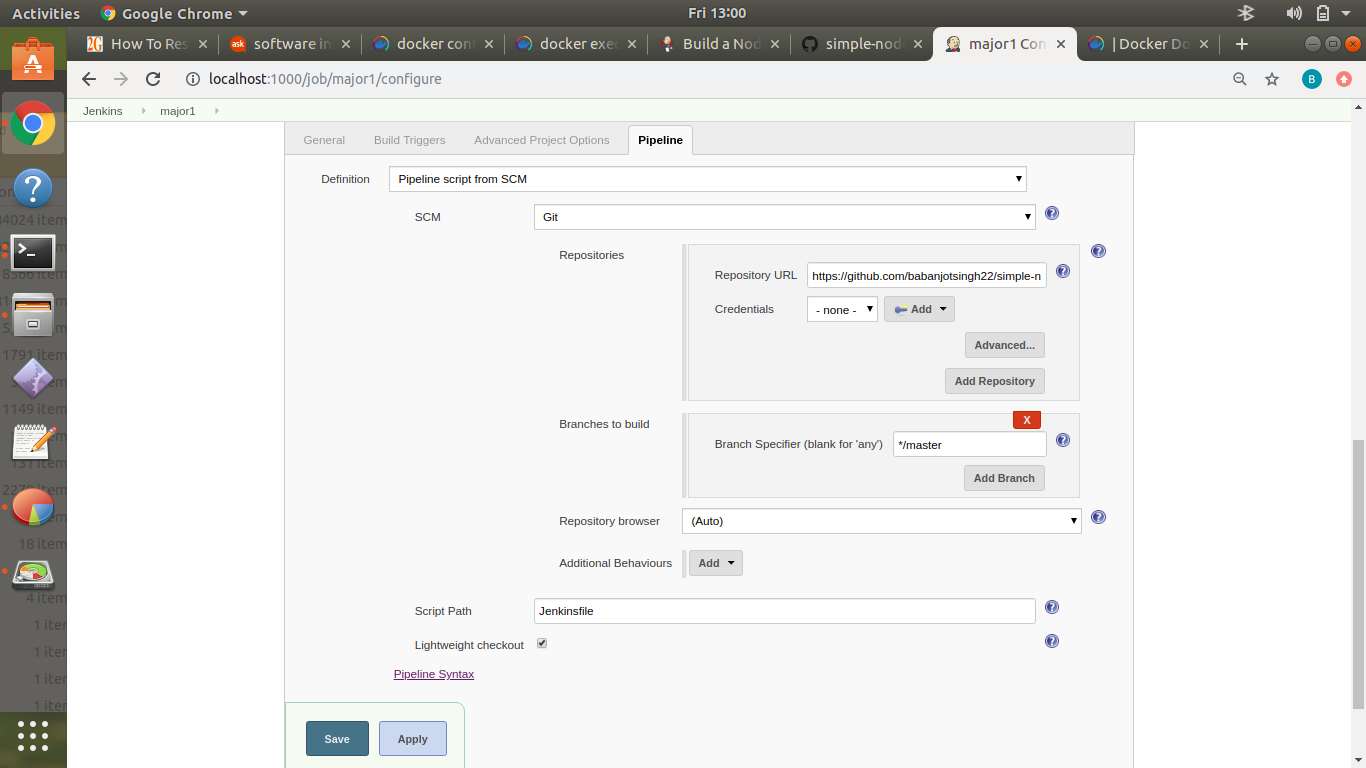


Figure 7.3.4.2 Choosing Pipeline Script from SCM

Step 5: Creating Initial Pipeline as JenkinsFile

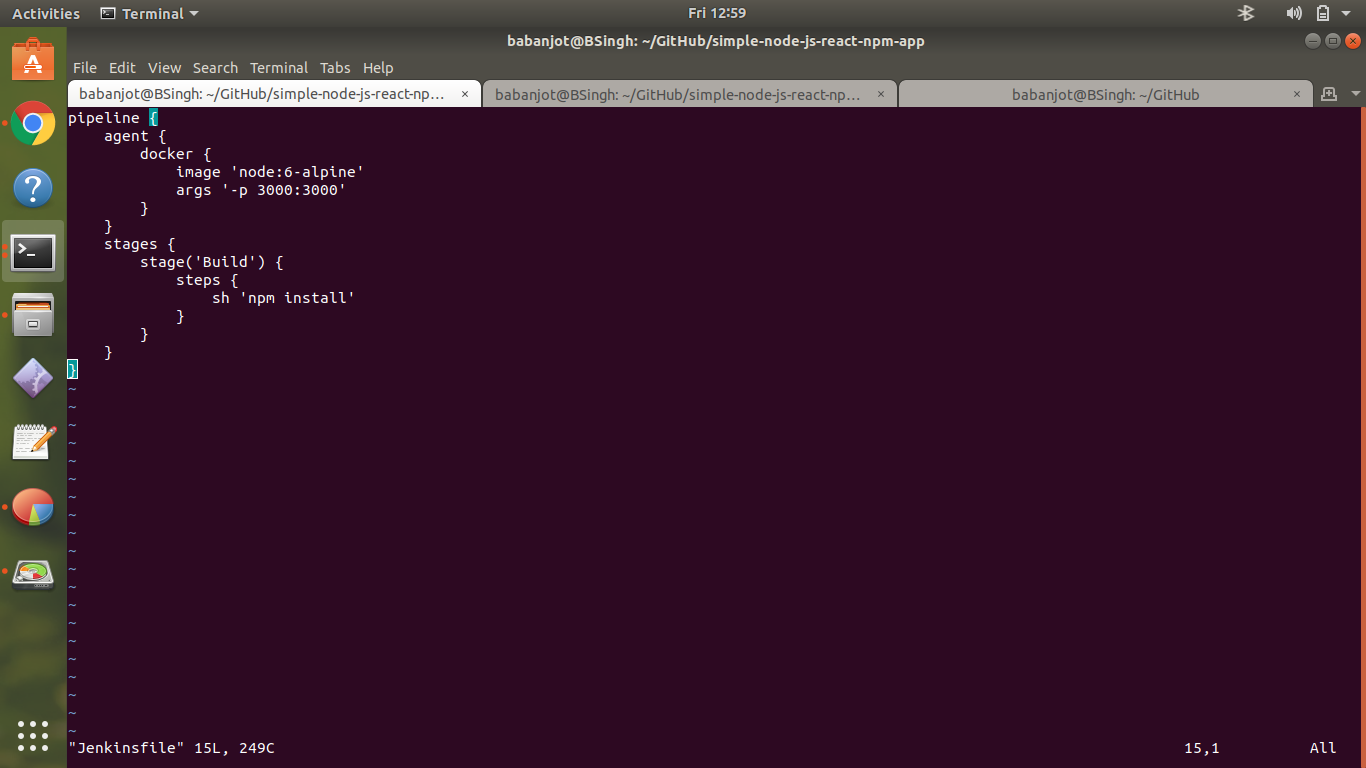


Figure 7.3.5 JenkinsFile

1. **Design**

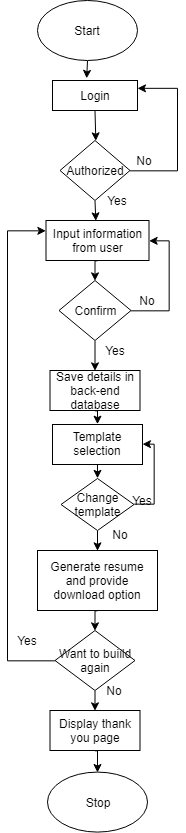
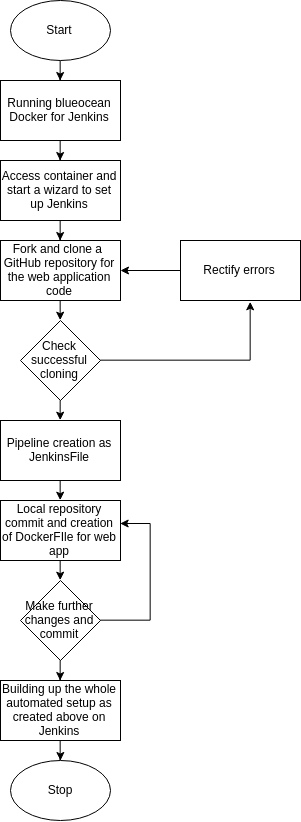
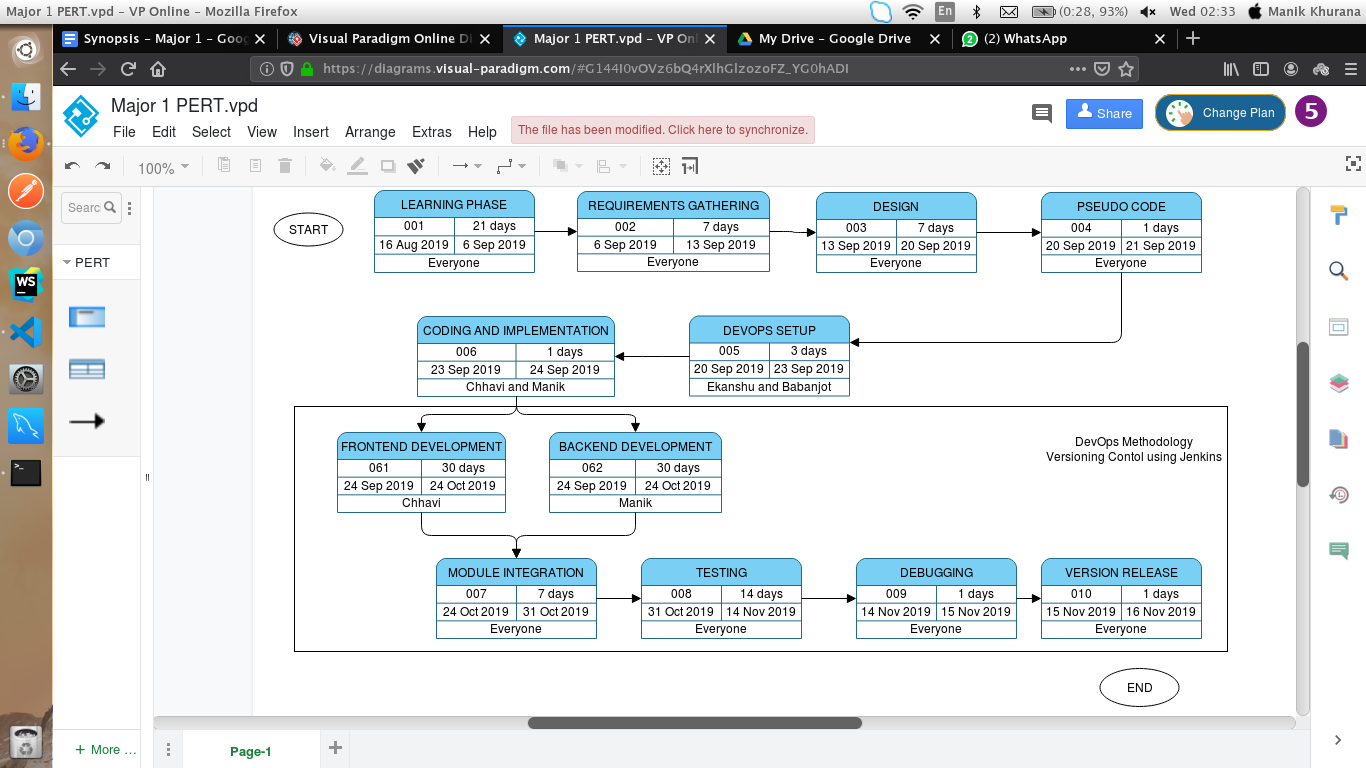
** **

Fig 8.1. Creation of web application Fig 8.2. Deployment of web application using DevOps

1. **System Requirements (Software/Hardware)**

* **Hardware:**
  + Intel(R) Core(TM) i3-3200 CPU @ 1.5 GHz
  + 2 GB RAM
  + System type is 32/64-bit Operating System
* **Software:**
  + Linux based operating System
  + Docker
  + Jenkins

1. **Schedule** (**P**roject **E**valuation and **R**eview **T**echnique Chart**)**

****

1. **References**

[1] Dunlu PENG† , Lidong CAO, Wenjie XU “Using JSON for Data Exchanging in Web Service Applications ”,  *University of Shanghai for Science and Technology, Shanghai 200093, China* , 2011.

[2] Naimul Islam Naim,“ReactJS: An Open Source JavaScript Library for Front-end Developement” *IMetropolia University of Applied Sciences*,30 May 2017.

[3] Kenneth Lewenhagen, Anders Åkesson .“Node.js in Open Source projects on Github.” *Blekinge Institute of Technology*

[4]. Arpitha R & Mrs. Kavitha S N,“Automation Using Jenkins: Plugins,Test Design , Test Test Execution and Reporting” *University of Zurich, Zurich, Switzerland.*

[5]. Jurgen Cito, Vincenzo Ferme, and Harald C. Gall, “Using Docker Containers to ImproveReproducibility in Software and WebEngineering” *Dept. of ISE, R. V. College of Engineering, Bengaluru, India.*

[6]. Darshita Kalyani, Dr. Devarshi Mehta,“Paper on Searching and Indexing Using Elasticsearch” *International Journal Of Engineering And Computer Science ISSN:2319-7242*

[7]. Kris Shaffer ,“Push, Pull, Fork: GitHub for Academics”, May 26, 2013

Links:

[a].[https://nodejs.org/en/doc](https://nodejs.org/en/docs/)/

[b].<https://reactjs.org/docs/getting-started.html>

[c].<https://jenkins.io/doc/>

[d].[https://docs.docker.com](https://docs.docker.com/)

[e].[https://docs.aws.amazon.com](https://docs.aws.amazon.com/)

[f]. [https://guides.github.com](https://guides.github.com/)

[g].[https://www.elastic.co/guide](https://www.elastic.co/guide/index.html)