International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 04 | Apr 2022 www.irjet.net p-ISSN: 2395-0072

Creation Of Social Group in Full Stack Progressive Apparel Purchasing Web App

Aniket Patil[1], Pritam Rane[2], Akash Khatkale[3], Sushopti Gawade[4]

[1-3]*Student, Dept. of Information Technology, Pillai College of Engineering, Navi Mumbai, India*
[4]*Professor, Dept. of Information Technology, Pillai College of Engineering, Navi Mumbai, India*

Abstract - *It is nearly hard for people to go out and purchase in these modern times, when there is very little time for people to go outside and shop. We devised the idea of shopping for garments from the comfort of one's own home using a web browser. Webstore selling clothing, shoes, and other wearable goods. Every product will be correctly classified so that you can easily navigate our website. A unique feature of our app will be the ability to connect with your friends and suggest one another within the app. The application also has a chatbot for user queries. We will utilize Google Sign in and Firebase to authorize users and provide security. Users will be able to make payments through our app. Who has read and write access to your database is determined by Firebase Realtime Database Security.*

Key Words: **ReactJS, Ecommerce, Chatbot, Firebase, Clothing, Social Group**

1. INTRODUCTION

Ecommerce has been growing swiftly since past few years. Ecommerce applications like Amazon, Flipkart are the biggest players in the game. Ecommerce has helped many small businesses grow and scale with quick pace. It has helped to bring customers and sellers closer by providing smooth and effortless services. Ecommerce is one of the biggest applications of how internet can be used efficiently. Small businesses are able to grow at a quick pace because of ecommerce. As the number of internet users grows, so does the number of online consumers, and many have been unable to go shopping for the past year and a half due to pandemics. A clothing purchase is also one of the fastest growing categories of virtual purchasing in India. According to a poll of garment shoppers in India, half of them shop online for better prices and a wider range of options. Almost 20% of buyers who use this as a payment method are concerned about online payment security (PwC Study, 2013). As a result, we're attempting to build a web store with a secure payment gateway. A website where you may purchase clothing, shoes, and other wearable goods. Our app will provide the ability to connect with friends and see their latest timeline (purchases) and wish list. There will be a section where a user can add his or her friends or acquaintance contact and then give recommendation other user in the chat section.

2. LITERATURE SURVEY

A. ReactJS: A Modern Web Development Framework [1]: ReactJS is used specifically for building UIs for single-page applications. ReactJS enables software developers to create large web-applications that can consume data and alter over a period of time without reloading the page. The use of reusable components allows us to develop our application in a straightforward manner.

B. Using Firebase Cloud Messaging to Control Mobile Applications [2]: FCM stands for (Firebase Cloud Messaging), and it is supported by Google. The suggested system acts as a Rest Client, sending information in JSON format with key-value pairs that is generated within the system or by using an existing Data message from the database.

C. Performance Optimization Techniques for ReactJS [3]: This paper discusses several practical methods for resolving such issues within the application, hence improving the performance of the ReactJS App in a production setting. In addition, the study will provide a time-efficient search technique for searching items in a big data collection by splitting the main component into individual components.

D. Study & Development of E-Commerce Website [4]: This paper describes how E-commerce websites can be designed and how they are bringing change in perspective of buyers as well as sellers. E-commerce means electronic commerce. E-commerce as we all know today is buying and selling of products and its services, or the transferring of money or data, over a system, mostly using the internet. E-commerce is improving the efficiency of the businesses that are being operated only using the Internet.

E. Web Chat using React Framework [5]: This study presents a chat-driven online system for enabling continuous collaboration among clients in a shared environment. The backend interface consists of two servers: client and server. The server side is in charge of database retrieval, database maintenance, and client-side services, whereas the client side is in charge of page maintenance and application interface. As middleware, reactjs, node, and json are used in this framework.

3. EXISTING METHODOLOGY

The existing system Architecture is shown in Fig. 1

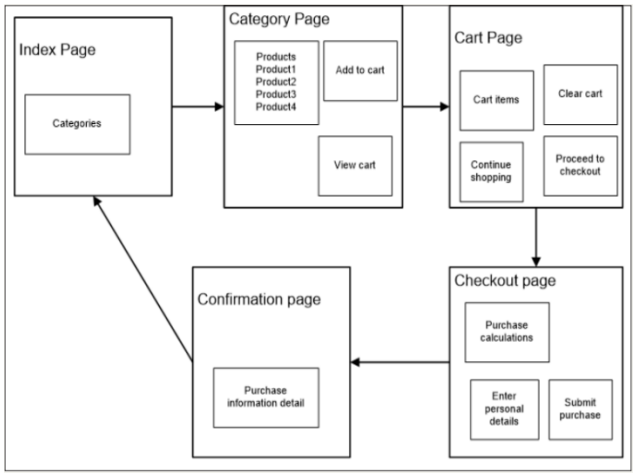

Fig-1: Existing System Architecture

The existing system architecture consists of static web pages which require loading each time the user wants to navigate to another page. Each Time the all the components of the page such as header and button elements were getting re rendered when page reloads.

4. PROPOSED WORK

We will be creating a web store with a secure payment gateway. A web Store where you can buy clothes, Shoes and other wearable accessories. A special feature of our app will be the Ability to connect with friends and see their recent timeline (purchases) and their wish list. There will be a section where a user can add his or her friends or acquaintance contact through email and then when any one of them makes a purchase the other user in the timeline can see what they have purchased.

4.1 Proposed System Architecture

In the proposed system architecture, we have tried to implement the technologies which help in efficiency of the application. The architecture implemented here is divided into three parts/subcategories.

i) User Interface

ii) Middleware

iii) Database

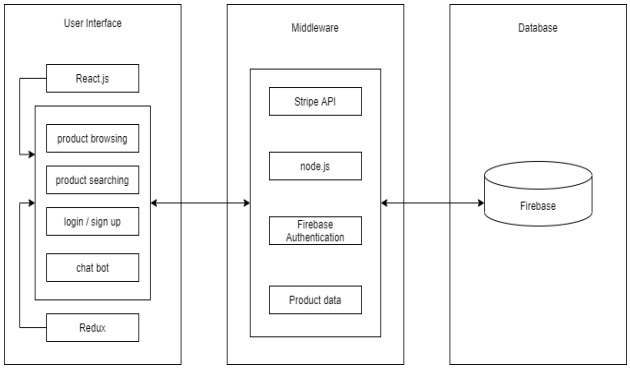

Fig-2: Proposed System Architecture

5. REQUIREMENT ANALYSIS

i) User Friendly Design: The User Interface must be easily graspable and convenient for the users to use.

ii) smooth and easy Checkout: The Checkout process is the steps a customer performs to purchase particular items. This includes every step that a customer performs from viewing the product to adding the product to cart to checking out the product. An ideal checkout flow will have a smaller number of steps to perform, a user-friendly design, and a smooth frictionless experience.

iii) Speed: This is achieved by reducing the memory used by the pages, making our server 24/7 available, using a content delivery network, caching memory and in memory or by compressing the data.

iv)Inbuilt Payment Methods: A smooth, error-free payment system will be integrated directly into the website. This process will eliminate any human interference and avoid possible human-errors.

v)Searching: To filter out the different products present in the website and show the appropriate filtered results to the users for efficient searching.

vi)Accessibility: Web accessibility refers to visual impairments that prevent users from using the Internet.

vii)Mobile Friendly: The website is user-friendly and responsive on mobile devices.

viii)Security: The data the user will provide to us will be secured.

5.2 Software

Our web application is built on ReactJS, Hooks, ContextAPI, Stripe and Firebase. react router and firebase will be used to have different pages on our app. We will add redux into our application so that it's ready to scale. We're also going to add payments using the stripe API to make payment on our store. We will use redux to store state. We will make

use of redux saga to handle asynchronous actions in our app. Software used will be visual studio code which supports all languages and helps better development of web applications. Also, for machine learning algorithms we will use jupyter notebook which uses python language, as our algorithms are based on python. We will be doing unit tests, integration tests and automation tests on our Software using Jasmine, Mocha, snapshot testing with JEST and Enzyme.

5.3 Hardware

Hardware requirements for our application are a computer or laptop with minimum configuration that could support visual studio code. For eg- a computer with 1 GB of RAM and 1.5 Ghz or faster processing speed.

6. IMPLEMENTATION

6.1 Overview

i) Software engineering: Software engineering is a systematic approach to software development that is based on engineering principles. A software engineer is someone who develops, designs, deploys, maintains, tests, and evaluates computer software using software development concepts. The term programmer is sometimes used interchangeably, but it does not necessarily mean engineering training or expertise. The software development process, including the definition, implementation, evaluation, measurement, management, modification, and improvement of the software lifecycle process itself, is supported by technical methods. Extensive use of software configuration management to handle systematic regulation of configuration changes to ensure configuration and code integrity and traceability throughout the system life cycle. Software version control tools like GitHub, SVN and so on are being used in modern workflows.

ii) Software testing: The process of validating and confirming the artifacts and behavior of the software under test cases performed by the user is known as software testing. Software testing can give a corporation an objective, unbiased view of the software, allowing them to understand and move forward with the risks associated with software implementation. Test techniques include:

a) A review of product requirements for integrity and accuracy in a variety of contexts, including industry, business, implementation feasibility and feasibility, ease of use, performance, security, and infrastructure issues

b) analyzing the architecture of the product along with the complete design

c) collaborating with product developers to improve codes efficiency, design patterns, and techniques such as boundary conditions tests that can be written as code

d) testing an application or program in order to see the behavior

e) studying the deployment architecture, as well as the code and automation that go along with the architecture

f) using visualizing and observability principles to take part in manufacturing activities

6.2 Implementation Details

The implementation of our proposed system is done using react.js which is a JavaScript library to create single page applications. The web application contains chatbot to help ease users for their queries. Also, for recommendations of products to one another a chat system has been developed in the application.

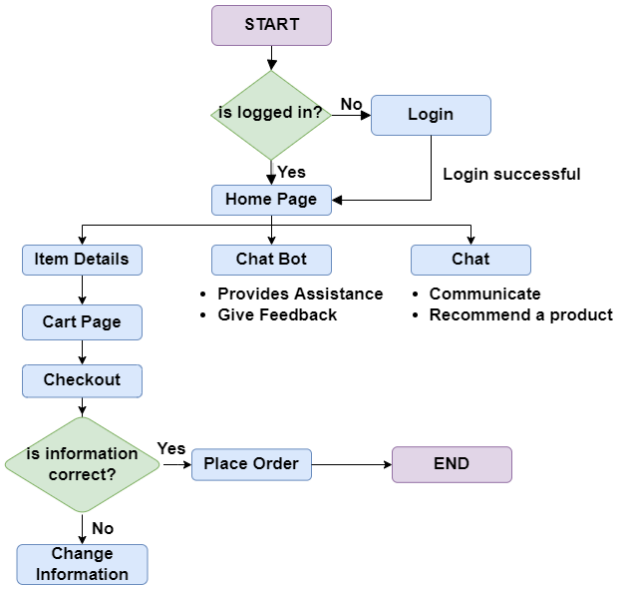

Fig-3: Implementation Plan

6.3 Methods/Techniques used:

1. Components: React.js divides the web page into several components because it is component-based. Each component is an element of the user interface design and has its own logic and appearance. As a result, component logic stated in JavaScript is simple, quick to execute, and reusable.

2. One-way data binding: As the name implies, one-way flow of the application is known as one-way data binding. In most cases, data in react flows only in a single direction: either from top to bottom or from parent to child components. The attributes (props) of a child component cannot return data to its parent component, but they can

communicate with the parent component to change the state on the basis of the inputs. One-way data binding operates in this manner. This keeps everything neat and tidy.

3. Performance: ReactJS is a quick framework. Because it is applied to VDOM(Virtual Document Object Model), the web application can run considerably faster than other front-end frameworks. React breaks down the complex user interface into discrete pieces. Multiple people can work on each element at the same time to shorten development time

6.4 Testing

Testing is the most crucial part of any software development process lifecycle. The process of validating and confirming the artifacts and behavior of the software under test cases performed by the user is known as software testing. The advantages of testing include the prevention of bugs, the reduction of development expenses, and the improvement of performance. Following flowchart depicts the types of testing methods and their functions.

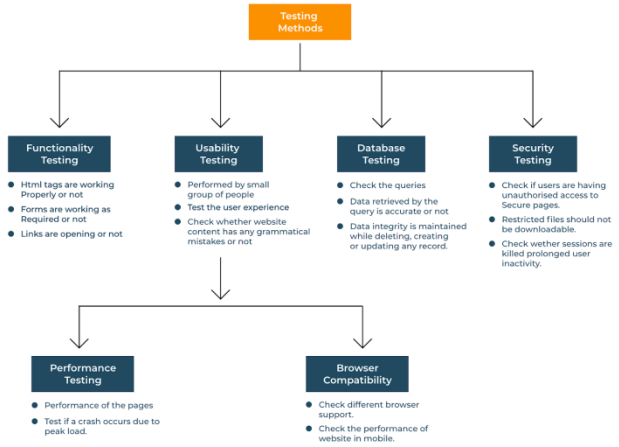

Fig-4: Testing Methods

Various testing methods used are:

1. Functional testing: In this we check if the HTML tags, forms and links are working properly or not.

2. Usability testing: This is performed by a small group of people to test the user experience.

3. Database testing: This testing is performed to validate the database queries.

4. Security testing: Used to check the security of the website, whether unauthorized users are able to access secure pages or not.

The testing method used in the implementation is 'snapshot testing'. A sort of "output comparison" or "golden master" testing is snapshot testing. Snapshot tests are simple to write and maintain, and they're a wonderful method to ensure that your app's behavior doesn't change abruptly while it's being developed. A snapshot test based on images would detect the change and highlight it.

The frameworks used here are Jest and enzyme.

1. Jest: Jest is a JavaScript testing framework that ensures the integrity of any JavaScript codebase. It enables you to develop tests with an easy-to-use, familiar, and feature-rich API that returns results rapidly.

2. Enzyme: Enzyme is a React JavaScript Testing utility tool that makes it easy to highlight, change, and explore the output of your React Components.

Jest and Enzyme go hand in hand with each other where Jest is used as the test executor, highlighting library, and mocking library and Enzyme is used to provide additional testing methods to interact with elements.

6.5 Outputs

6.5.1 Index Page

This is the Home Page of the application which contains chat, sign in, chatbot shop and different product previews.

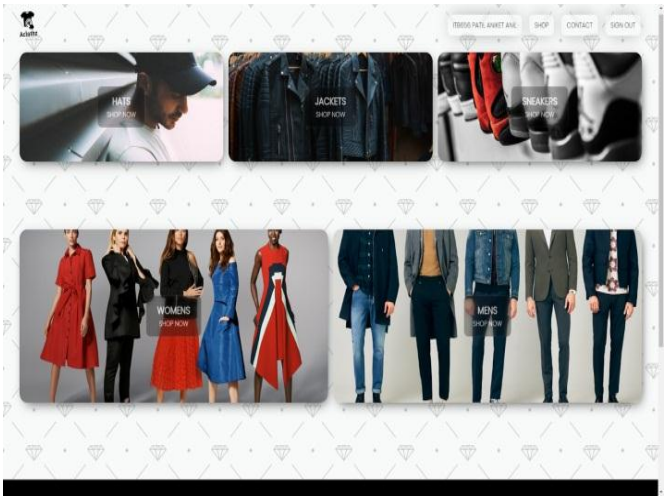

Fig-5: Index Page

6.5.2 Shop Page

Shop page contains the products of different categories respectively.

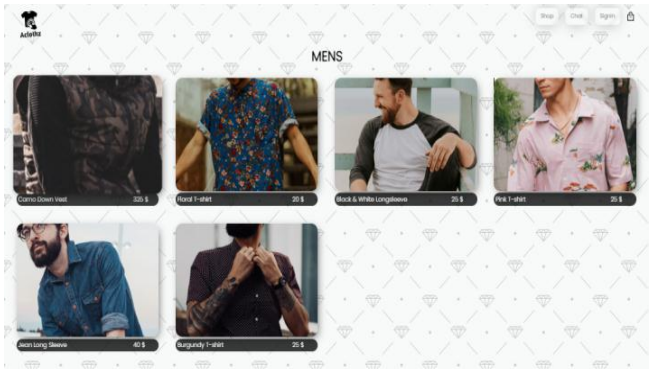

Fig-6: Shop Page

6.5.3 Cart Component

The cart component is dynamic and loads items added to cart without refreshing the page.

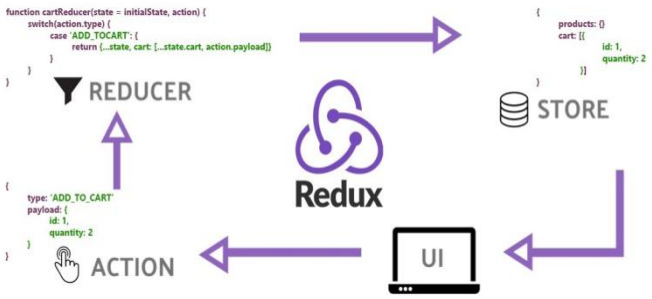

Fig-7: Cart Working

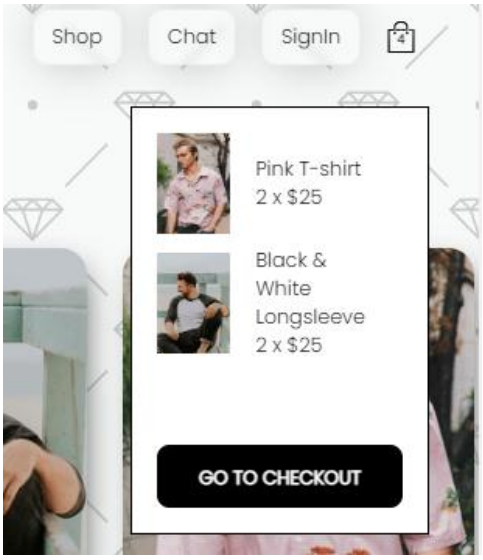

Fig-8: Cart

6.5.3 Chat Page

Here the users can socially interact with each other using the chat option.

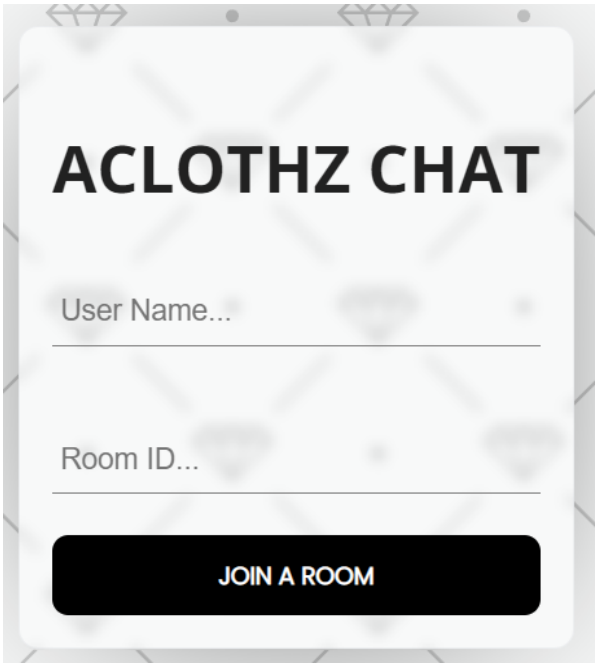

Fig-9: Chat Page

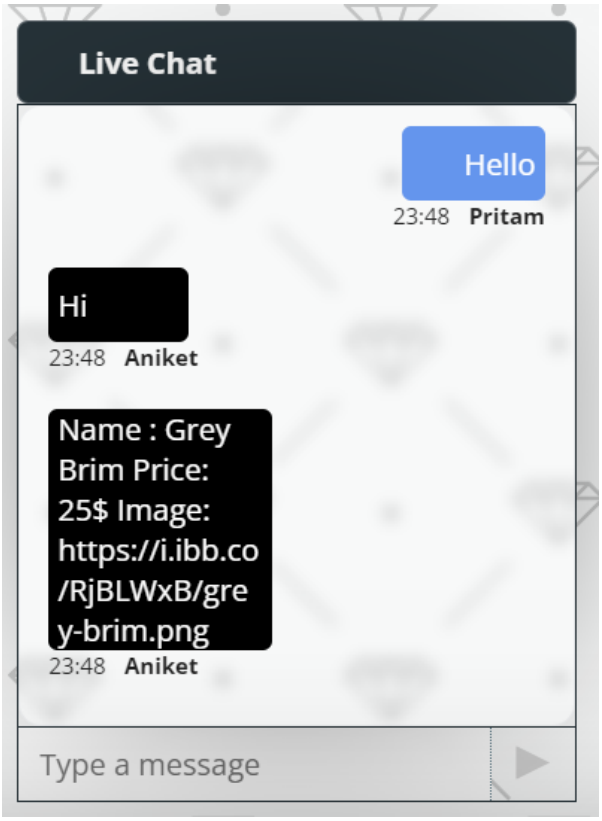

Fig-10: Chat Feature

6.5.4 Testing outputs

1. Snapshot Testing: A sort of "output comparison" or "golden master" testing is snapshot testing. A snapshot test based on images would detect the change and highlight it.

2. Benefits of snapshot: Writing test takes a lot of time and snapshot testing does testing and visualization of differences using shorter codes

```
it('should render collection-item component', () => {
  expect(wrapper).toMatchSnapshot();
});

it('should action add to cart when click', () => {
  wrapper.find(CustomButton).first().simulate('click');
  expect(mockDispatch).toHaveBeenCalledWith(addCartItem(mockCartItem));
})
});
```

```
PASS src/components/collection-item/collection-item.test.js
  collection-item compnent
    √ should render collection-item component (11 ms)
    √ should action add to cart when click (35 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   1 passed, 1 total
Time:        4.284 s
Ran all test suites matching /src\\components\\collection-item\\collection-item\.test\.js/i.

Watch Usage: Press w to show more.
```

Fig-11: Testing

7. FUTURE SCOPE

The scope of this project could be extended to following points. For the convenience of the user in order to make website more user friendly, Introduction of product recommendation based on the previous history of the user can be done. Improve website for faster loading and user convenience. We could also Implement a physical Store where there will be no physical cloths available but there will be virtual changing rooms where an advanced computer will be setup with 3D scanning hardware and Augmented reality enabled to make the exact model of the person standing in front of the screen and that person can choose any outfit that suits them and order it immediately without needing to look for which size fits them the best as the 3D scanning hardware will automatically calculate the best fit. This way there will be no physical contact with anyone and there will be no need for a salesperson inside the store.

8. SUMMARY

In this report we have explained the domain of our project, which is a full stack progressive apparel purchasing web app. We have described the implementation details of the project, all the software and hardware requirements for it, and the libraries and tools which will be used to develop the application. We have explained all the literature survey papers of all the research papers we used for reference to build the web application. We have also explained various testing methods we have done to test the efficiency of our application and also mentioned the frameworks we used for testing. The system architecture and the future scope of the project is also well explained.

REFERENCES

[1] Prateek Rawat, "ReactJS: A Modern Web Development Framework", International Journal of Innovative Science and Research Technology Volume 5, Issue 11, November – 2020

[2] Mohamed Abdalla Mokar, Sallam Osman Fageeri and Saif Eldin Fattoh, "Using Firebase Cloud Messaging to Control Mobile Applications", IEEE July-2019

[3] Arshad Javeed, "Performance Optimization Techniques for ReactJS", IEEE 2019

[4] Aftab Aalam, "Study & Development of E-commerce Website", IRJET May-2020

[5] Akhilesh Sarjit M S "Web Chat using React Framework", IJTSRD April-2020

[6] Piotr Roksela, Marek Konieczny and Slawomir Zielinski, "Evaluating execution strategies of GraphQL queries", IEEE 2020.

[7] Kanakamedala Deepika, Subetha T, Veeranki Tilekya and Jatroth Mamatha, "Jollity Chatbot- A contextual AI Assistant", Proceedings of the Third International Conference on Smart Systems and Inventive Technology (ICSSIT 2020) IEEE 2020 Part Number: CFP20P17-ART; ISBN: 978-1-7281-5821-1

[8]React Docs - https://reactjs.org/docs/getting-started.html

[9] Firebase Docs - https://firebase.google.com/docs

[10] Nodejs Docs - https://nodejs.org/en/docs/