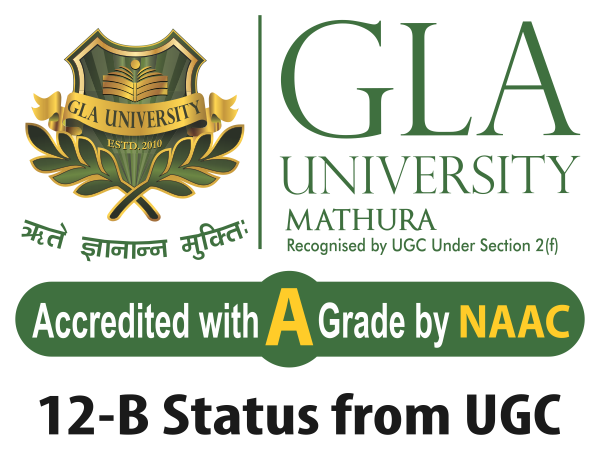
FullStack Project Report

(2020-2021)



**Amazon Clone**

**Institute of Engineering and Technology**

Submitted by :

Kanishk Bharadwaj (181500307)

Ananya Jain (181500091)

Akansha Saxena (181500054)

**Declaration**

We hereby declare that the work which is being presented in Full Stack Project **“**Amazon Clone**”,** in partial fulfilment of the requirements for FullStack viva voce, is an authentic record of our own work carried under the supervision of **Mr. Pankaj Kapoor, Assistant Professor, GLA University, Mathura.**

Ananya Jain (181500091)

Sign : \_\_\_\_\_\_\_\_\_\_\_\_\_

Akansha Saxena (181500054)

Sign : \_\_\_\_\_\_\_\_\_\_\_\_\_

Kanishk Bharadwaj (181500307)

Sign : \_\_\_\_\_\_\_\_\_\_\_\_\_

Course: B.Tech(CSE)

Year: 3rd

Semester: VI

Members Github Id’s :

<https://github.com/Ananya015>

https://github.com/Akansha0211

<https://github.com/coderkabh>

**Certificate**

This is to certify that the project entitled “**Amazon Clone**” carried out in FullStack Project is the work done by Ananya Jain , Akansha Saxena, Kaniskh Bharadwaj and is submitted in partial fulfilment of the requirements for the award of degree Bachelor of Technology (Computer Science and Engineering).

Signature of Supervisor:

Name of Supervisor: **Mr. Pankaj Kapoor**

Date:

|  |  |
| --- | --- |
|  | **Acknowledgement**    It is our pleasure to acknowledge the assistance of a number of people without them this project would not have been possible.  First and foremost, we would like to express our gratitude to **Mr. Pankaj Kapoor,** our project mentor, for providing invaluable Encouragement, guidance and assistance. We would like to thank my co-team members for their complete support throughout in finishing the mentioned project accurately. After doing this project we can confidently say that this experience has not only enriched us with technical knowledge but also has unparsed the maturity of thought and vision, the attributes required for being a professional.  **Abstract:** We see a lot of projects in resumes which are made by students but many of them do not have a backend and even if they have then the implementation is not proper. In this project we (Akansha, Ananya and Kanishk) have made a very attractive and simple website which is a clone of an E-Commerce website we use in our daily life i.e. Amazon.in.  We have made a clone of Amazon Shopping website which is smooth in design, as cool as and as good as Amazon with all features of Amazon shopping website. It has the UI and dynamic features which the website Amazon has.  **Table of Contents**  **Declaration……………………………………………………… 2**  **Certificate……...…………...…………………………………… 3**  **Acknowledgement…………..………………………………….. 4**  **Abstract…………………….…………………………………… 5**  **Table of Content………………………………………………… 6-7**  **1. Introduction**  1.1 Overview………………..……………..………………………………….... 8  1.2 Motivation….…………………………....…………………………….……8  1.3 ProjectPlan…………………………………….……………………………….8  1.3.1 Objective  1.3.2 Scope  1.4 Drawbacks in existing system………..………..…………………………….9  **2. Software**  2.1 Hardware Requirements…………………………….………………………. 10  2.2 Software Requirements……………………………….…………………….. 10  2.3 Installation of VS Code……………………………………………………. 11  2.4 Specific Requirements………………………………………………………..13  2.4.1 Language Used……………………………………………………………..13  **3. Software Design**  3.1 Use Case Diagram………………………………………………………..….. 16  3.2 Data Flow Diagram……………………………..…………………………… 16  3.3 Activity Diagram……………………………………………………………..17  **4. Testing**  4.1 Introduction.……………………………………………………….………….18  4.2 Objectives of Software Testing…..…….…………………………..……….18  4.2.1 Software Quality Improvement.……………………………..….…..……..19  4.2.2 Verification and Validation…………………………………………..……...19  **5. Implementation and User Interface……………………………………..19-39**  **6. References/Bibliography…………..…..…………………………………..40**  **7. Links…………………………………………………………………………..40**  7.1 Project GitHub Link…………………………………………………………40  7.2 Live Project Link…………………………………………………………….40 |
|  |  |

**Chapter 1**

**Introduction:**

**1.1 Overview:** This is an amazon clone which gives a dummy idea of that website. It has several features like authenticating the user by the use of firebase authentication. We have also implemented the cart with real-time access which has features like deleting and adding items into the cart and many more features are there.

We used ReactJS (which is a library of javascript) for the UI implementation of our project. We used cards, NavLink, cart, navbar, footer and many other UI components by using the features of React.Js such as single page website etc. We used this technology because it is easy to mount and unmount the components by the use of React.

For the backend part we have used Express.Js which is a framework of Node.Js which is used to handle server requests and to serve the web pages according to the need of the client. It is very easy to implement and also very reliable.

**1.2 Motivation:** The motivation for doing this project is to learn the implementation of ReactJS, ExpressJs with firebase authentication and other technologies so that we will learn something new.

**1.3.** **Project Plan**

**1.3.1. Objective**

The main objective of this project is to create an ecommerce website which will work as a single page website and would be able to give us the look and features of Amazon.

**1.3.2. Scope**

The scope of this project is to create a website as our startup which will place orders in future.

**1.4. Drawbacks in Existing System**

* No database uptill now
* No separate page for every product
* No payment gateway implemented

**Chapter-2**

**Software Requirement Analysis**

**2.1. Hardware Requirements**

* · **Processor : ·** Intel CORE i5 (8th Gen)
* · **Main Memory (RAM) :** 8GB RAM
* · **Monitor :** 14 inch color monitor
* · **Keyboard :** 108 Keys
* · **Mouse :** optical mouse
* · **Hard Disk :** 160GB

**2.2. Software Requirements**

· **System Software**

**- Operating System :** Windows 10 , Linux

· **Application Software**

- **Tools :** GitHub , VS Code, webStorm

- **Front-end :** React.Js

- **Back-end :** Node.Js and Express.Js

- **Other Technologies :** Firebase, Firebase Authentication. Firebase Hosting

**2.3. Installation of VS Code and WebStorm**

VS Code is a free code editor, which runs on the macOS, Linux, and Windows operating systems.VS Code is lightweight and should run on most available hardware and platform versions. You can review the [System Requirements](https://code.visualstudio.com/docs/supporting/requirements) to check if your computer configuration is supported.

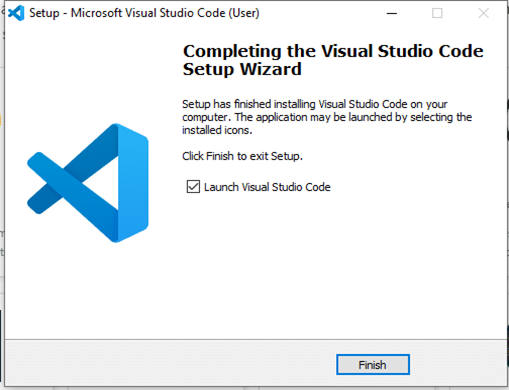


Fig 1. Installation of VS Code editor

**WebStorm** is a powerful IDE for modern JavaScript development. **WebStorm** provides full support for JavaScript, TypeScript, HTML, CSS as well as for frameworks such as React, Angular, and Vue. js right out of the box, no additional plugins are required

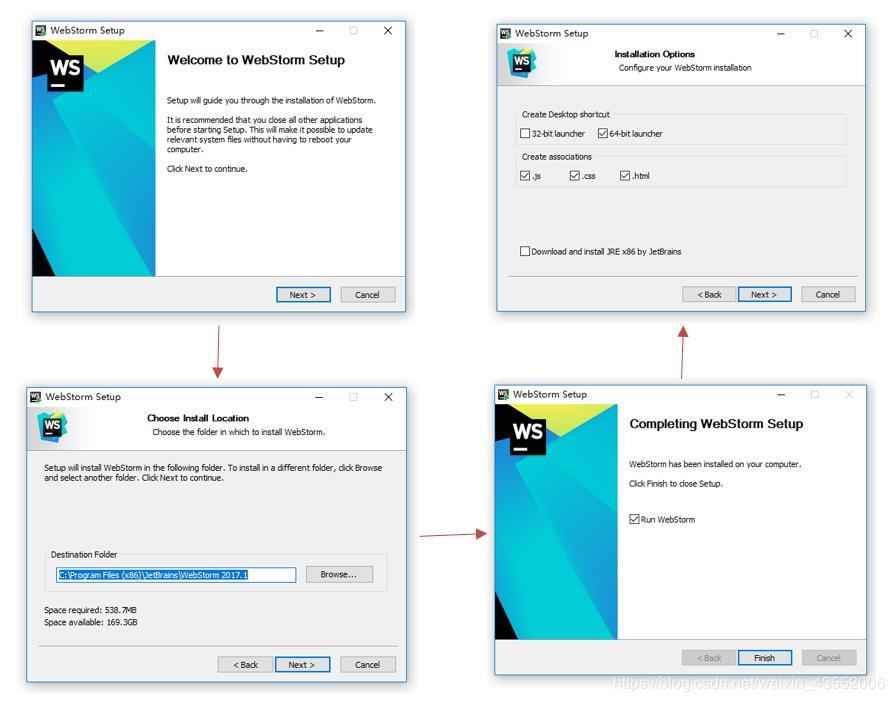


Fig 2. Installation of webstorm

Both vs code and webstorm are good for making frontend.

**2.4. Specific Requirements**

**2.4.1 Languages Used**

**HTML:**

**HTML** stands for **Hyper Text Mark-up Language**, which is the most widely used language on the Web to develop web pages. HTMLwas created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though the HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

Some of the key advantages of learning HTML:

• **Create Web site** - You can create a website or customize an existing web template if you know HTML well.

• **Understand web** - If you want to optimize your website, to boost its speed and performance, it is good to know HTML to yield best results.

• **Learn other languages** - Once you understand the basics of HTML then other related technologies like java script, php, or angular are easier to understand.

• **Become a web designer** - If you want to start a career as a professional web designer, HTML and CSS designing is a must skill.

**CSS :**

**C**ascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the colour of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the mark-up languages HTML or XHTML.

• **CSS saves time** − You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

• **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.

• **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

• **Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

**JavaScript :**

**JavaScript** is a lightweight, interpreted programminglanguage. It is designed for creating network-centric applications. It is complementary to and integrated with Java. JavaScriptis very easy to implement because it is integrated with HTML. It is open and cross-platform.

**ReactJS :**

React (also known as React.js or ReactJS) is an open-source, front end, JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies.React can be used as a base in the development of [single-page](https://en.wikipedia.org/wiki/Single-page_application) or mobile applications. However, React is only concerned with state management and rendering that state to the [DOM](https://en.wikipedia.org/wiki/Document_Object_Model), so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

Declarative views make your code more predictable and easier to debug.

Build encapsulated components that manage their own state, then compose them to make complex UIs.

Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

We don’t make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code.

React can also render on the server using Node and power mobile apps using React Native.

**Installing ReactJS using webpack and babel**

Webpack is a module bundler (manages and loads independent modules). It takes dependent modules and compiles them to a single (file) bundle. You can use this bundle while developing apps using the command line or, by configuring it using webpack.config file.

Babel is a JavaScript compiler and transpiler. It is used to convert one source code to another. Using this you will be able to use the new ES6 features in your code where babel converts it into plain old ES5 which can be run on all browsers.

Step 1 - Create the Root Folder

Create a folder with name reactApp on the desktop to install all the required files, using the mkdir command.

C:\Users\username\Desktop>mkdir reactApp

C:\Users\username\Desktop>cd reactApp

To create any module, it is required to generate the package.json file. Therefore, after Creating the folder, we need to create a package.json file. To do so you need to run the npm init command from the command prompt.

C:\Users\username\Desktop\reactApp>npm init

This command asks information about the module such as packagename, description, author etc. you can skip these using the –y option.

C:\Users\username\Desktop\reactApp>npm init -y

Wrote to C:\reactApp\package.json:

{

"name": "reactApp",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"keywords": [],

"author": "",

"license": "ISC"

}

Step 2 - install React and react dom

Since our main task is to install ReactJS, install it, and its dom packages, using install react and react-dom commands of npm respectively. You can add the packages we install to the package.json file using the --save option.

C:\Users\Tutorialspoint\Desktop\reactApp>npm install react --save

C:\Users\Tutorialspoint\Desktop\reactApp>npm install react-dom --save

Or, you can install all of them in single command as −

C:\Users\username\Desktop\reactApp>npm install react react-dom --save

Step 3 - Install webpack

Since we are using webpack to generate bundler install webpack, webpack-dev-server and webpack-cli.

C:\Users\username\Desktop\reactApp>npm install webpack --save

C:\Users\username\Desktop\reactApp>npm install webpack-dev-server --save

C:\Users\username\Desktop\reactApp>npm install webpack-cli --save

Or, you can install all of them in single command as −

C:\Users\username\Desktop\reactApp>npm install webpack webpack-dev-server webpack-cli --save

Step 4 - Install babel

Install babel, and its plugins babel-core, babel-loader, babel-preset-env, babel-preset-react and, html-webpack-plugin

C:\Users\username\Desktop\reactApp>npm install babel-core --save-dev

C:\Users\username\Desktop\reactApp>npm install babel-loader --save-dev

C:\Users\username\Desktop\reactApp>npm install babel-preset-env --save-dev

C:\Users\username\Desktop\reactApp>npm install babel-preset-react --save-dev

C:\Users\username\Desktop\reactApp>npm install html-webpack-plugin --save-dev

Or, you can install all of them in single command as −

C:\Users\username\Desktop\reactApp>npm install babel-core babel-loader babel-preset-env

babel-preset-react html-webpack-plugin --save-dev

Step 5 - Create the Files

To complete the installation, we need to create certain files namely, index.html, App.js, main.js, webpack.config.js and, .babelrc. You can create these files manually or, using command prompt.

C:\Users\username\Desktop\reactApp>type nul > index.html

C:\Users\username\Desktop\reactApp>type nul > App.js

C:\Users\username\Desktop\reactApp>type nul > main.js

C:\Users\username\Desktop\reactApp>type nul > webpack.config.js

C:\Users\username\Desktop\reactApp>type nul > .babelrc

**ExpressJS :**

Express.js, or simply Express, is a [back end](https://en.wikipedia.org/wiki/Front_end_and_back_end) [web application framework](https://en.wikipedia.org/wiki/Web_application_framework) for [Node.js](https://en.wikipedia.org/wiki/Node.js), released as [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) under the [MIT License](https://en.wikipedia.org/wiki/MIT_License). It is designed for building [web applications](https://en.wikipedia.org/wiki/Web_application) and [APIs](https://en.wikipedia.org/wiki/API). It has been called the [de facto standard](https://en.wikipedia.org/wiki/De_facto_standard) server framework for [Node.js](https://en.wikipedia.org/wiki/Node.js).

Express is the back-end component of popular development stacks like the [MEAN](https://en.wikipedia.org/wiki/MEAN_(software_bundle)), [MERN](https://en.wikipedia.org/w/index.php?title=MERN&action=edit&redlink=1) or [MEVN](https://en.wikipedia.org/w/index.php?title=MEVN&action=edit&redlink=1) stack, together with the [MongoDB](https://en.wikipedia.org/wiki/MongoDB) database software and a [JavaScript](https://en.wikipedia.org/wiki/JavaScript) front-end framework or library.

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

With a myriad of HTTP utility methods and middleware at your disposal, creating a robust API is quick and easy.

Express provides a thin layer of fundamental web application features, without obscuring Node.js features that you know and love.

Many popular frameworks are based on Express.

Assuming you’ve already installed Node.js, create a directory to hold your application, and make that your working directory.

**Installation:**

$ mkdir myapp

$ cd myapp

Use the npm init command to create a package.json file for your application. For more information on how package.json works, see Specifics of npm’s package.json handling.

$ npm init

This command prompts you for a number of things, such as the name and version of your application. For now, you can simply hit RETURN to accept the defaults for most of them, with the following exception:

entry point: (index.js)

Enter app.js, or whatever you want the name of the main file to be. If you want it to be index.js, hit RETURN to accept the suggested default file name.

Now install Express in the myapp directory and save it in the dependencies list. For example:

$ npm install express --save

To install Express temporarily and not add it to the dependencies list:

$ npm install express --no-save

**Firebase authentication :**

Firebase Authentication aims to make building secure authentication systems easy, while improving the sign-in and onboarding experience for end users. It provides an end-to-end identity solution, supporting email and password accounts, phone auth, and Google, Twitter, Facebook, and GitHub login, and more.

FirebaseUI provides a customizable, open source, drop-in auth solution that handles the UI flows for signing in users. The FirebaseUI Auth component implements best practices for authentication.

Built by the same team that developed Google Sign-in, Smart Lock and Chrome Password Manager, Firebase security applies Google's internal expertise of managing one of the largest account databases in the world.

It can take months to set up your own auth system, and it requires an engineering team to maintain that system in the future. Set up the entire authentication system of your app in under 10 lines of code, even handling complex cases like account merging.

Firebase helps you develop high-quality apps, grow your user base, and earn more money. Each feature works independently, and they work even better together.

**Chapter-3**

**Software Designs**

**3.1 Use Case diagram**

A use case diagram is a dynamic or behaviour diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.

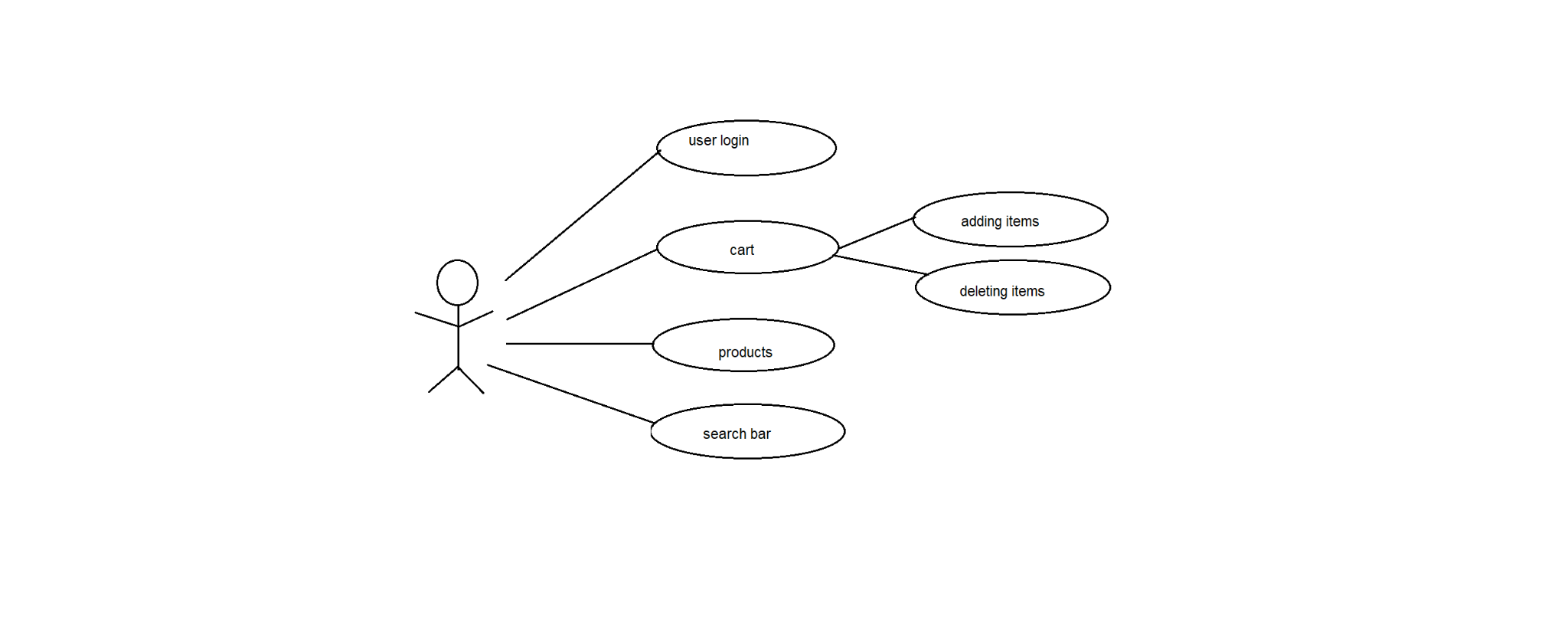
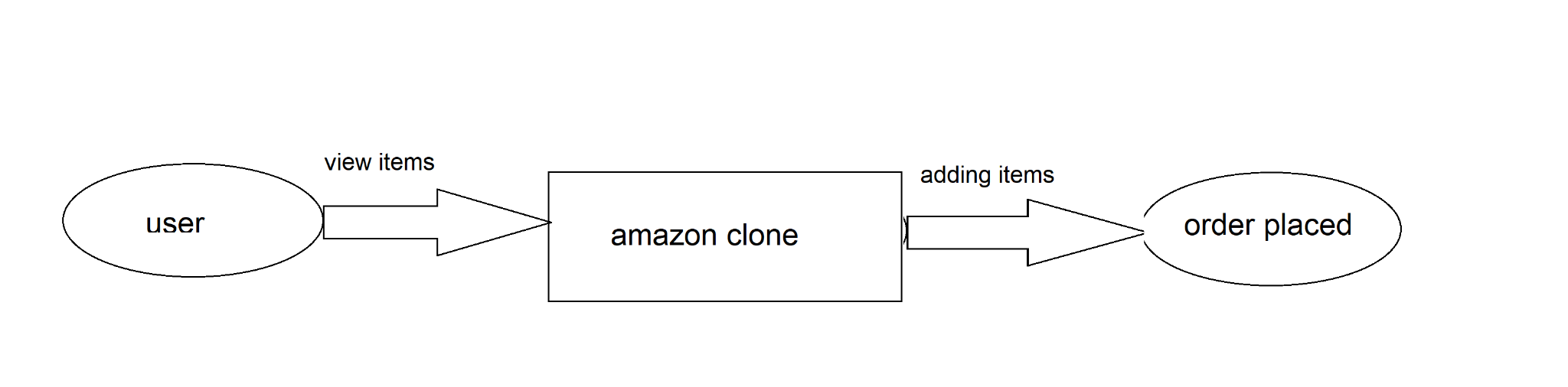
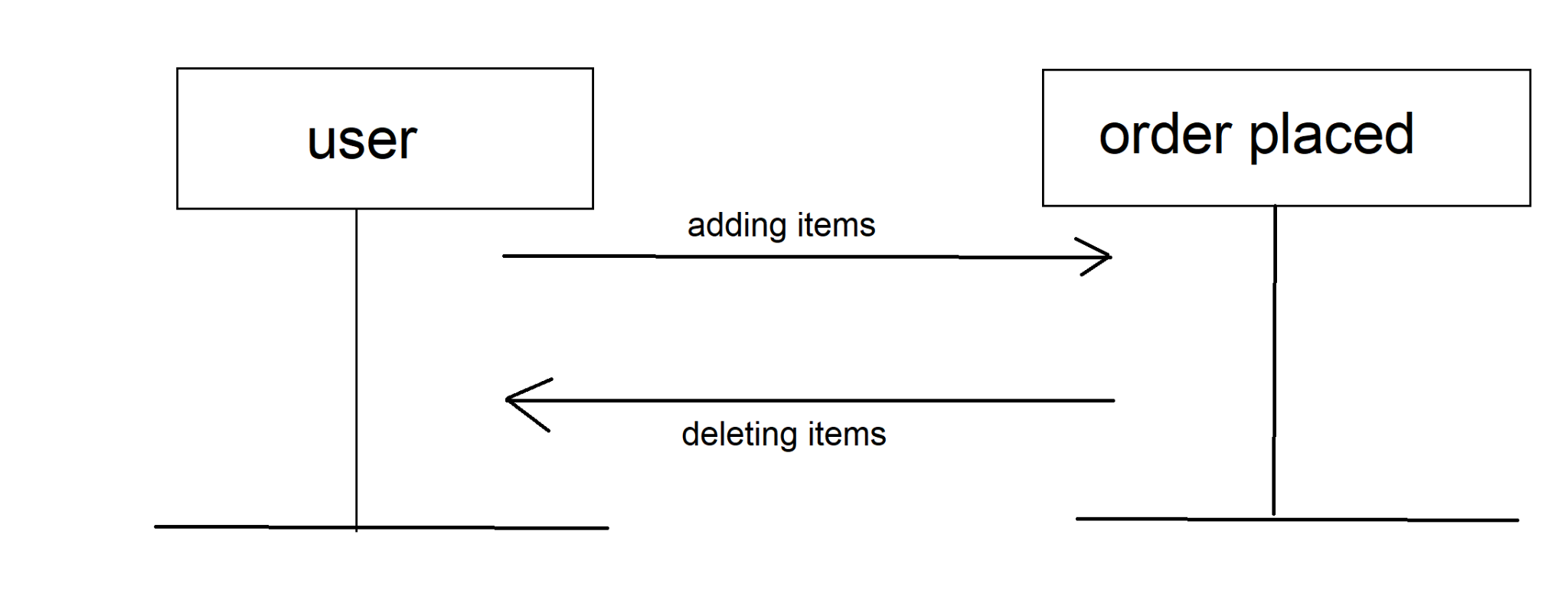


Fig 3.1 Use Case diagram

**3.2 Data Flow diagram  
Level 0:**

**Fig. 3.2Level-0 DFD**

**3.3 Activity diagram**



**Fig. 3.3 Sequential Diagram**

**Chapter-4**

**Testing**

**4.1 Introduction**

Software testing is the process of executing a program with the intention of finding errors in the code. It is a process of evolution of system or its parts by manual or automatic means to verify that it is satisfying specified or requirements or not.

The implementation phase of software development is concerned with translating design specification into source code. The preliminary goal of implementation is to write source code and internal documentation so that conformance of the code to its specifications can be easily verified, and so that debugging, testing and modifications are eased. This goal can be achieved by making the source code as clear and straightforward as possible. Simplicity, clarity and elegance are the hallmark of good programs, obscurity, cleverness, and complexity are indications of inadequate design and misdirected thinking.

## 

## 

## 

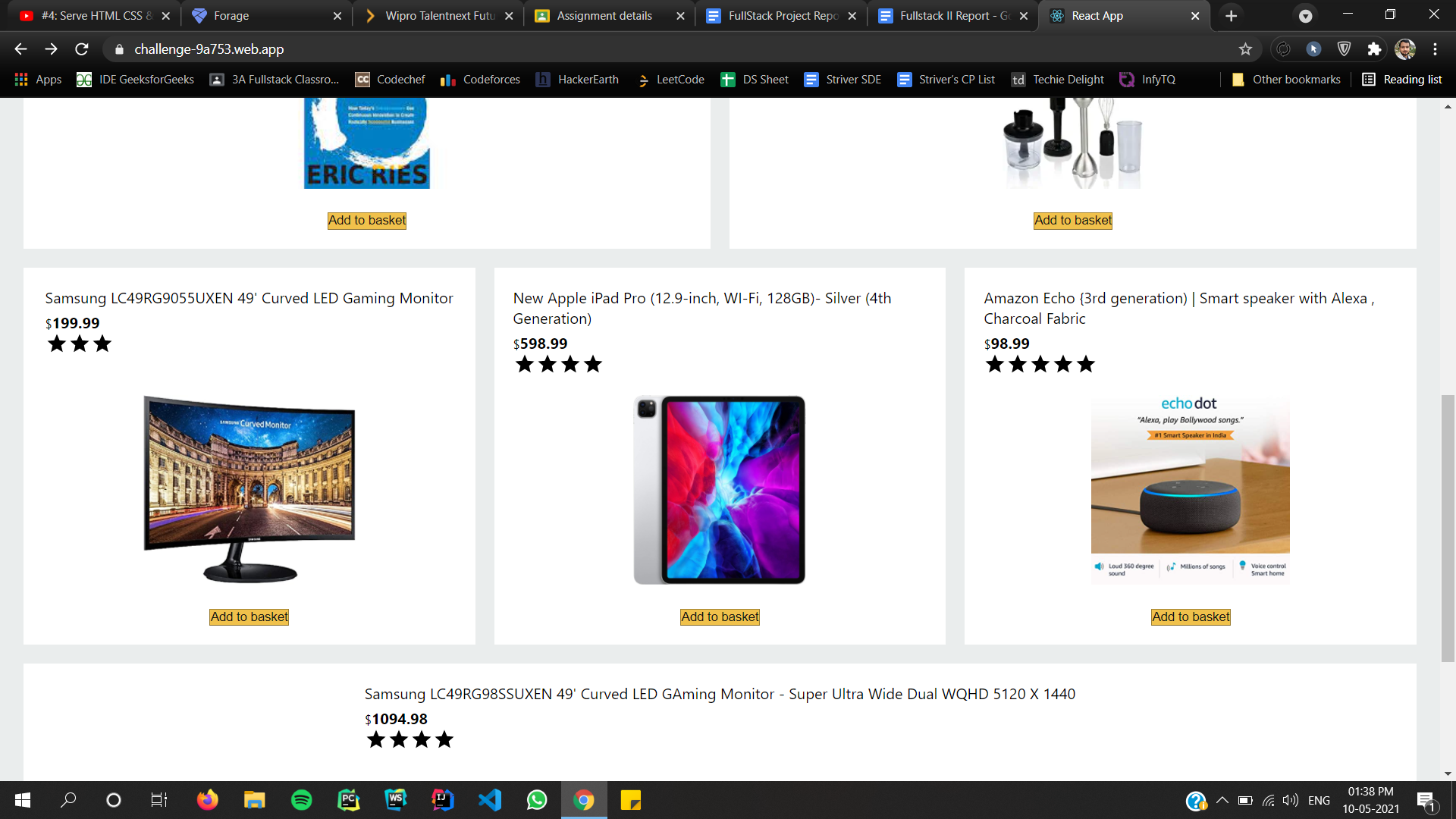
## **4.2** **Objectives of Software Testing**

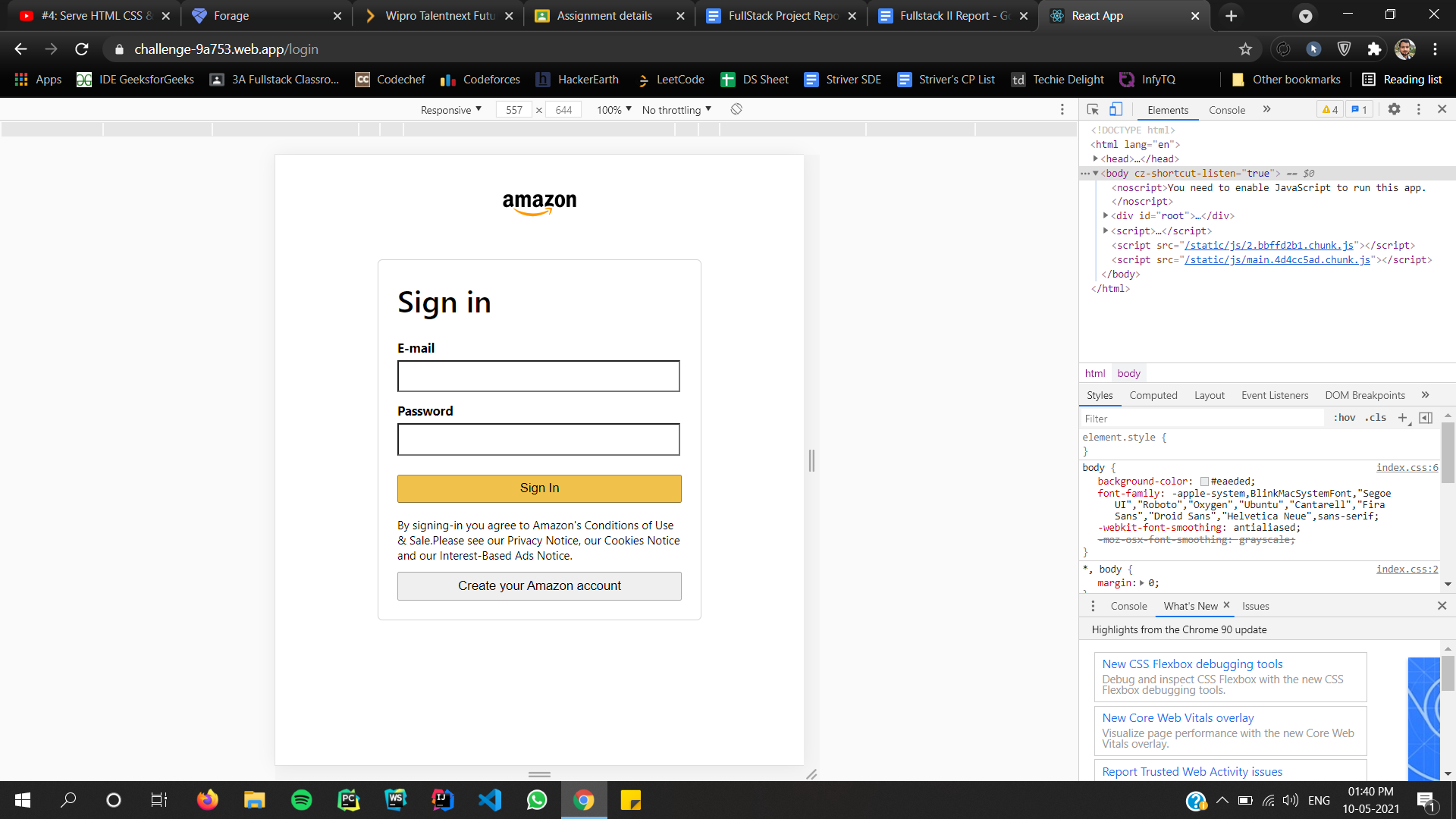
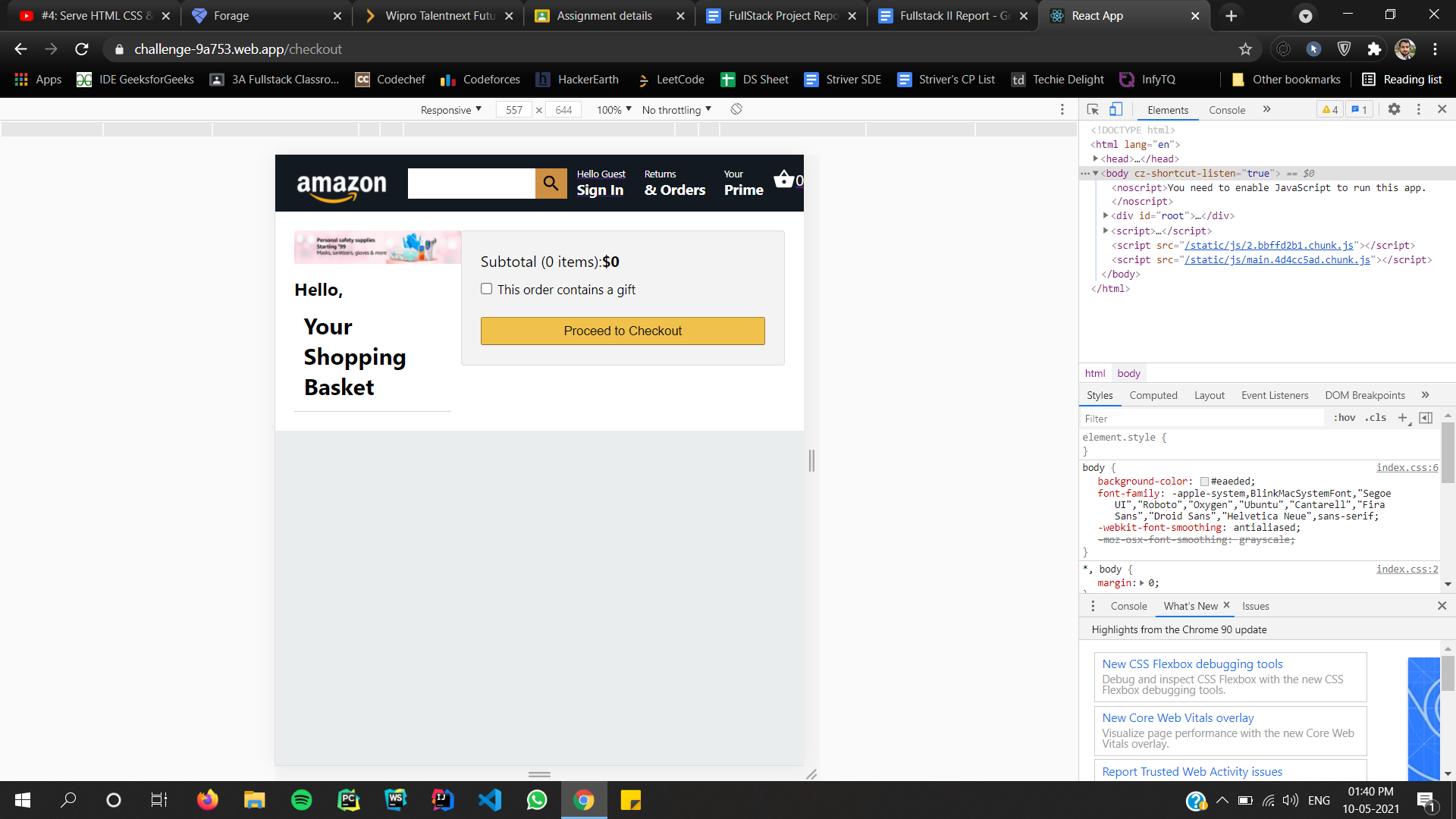
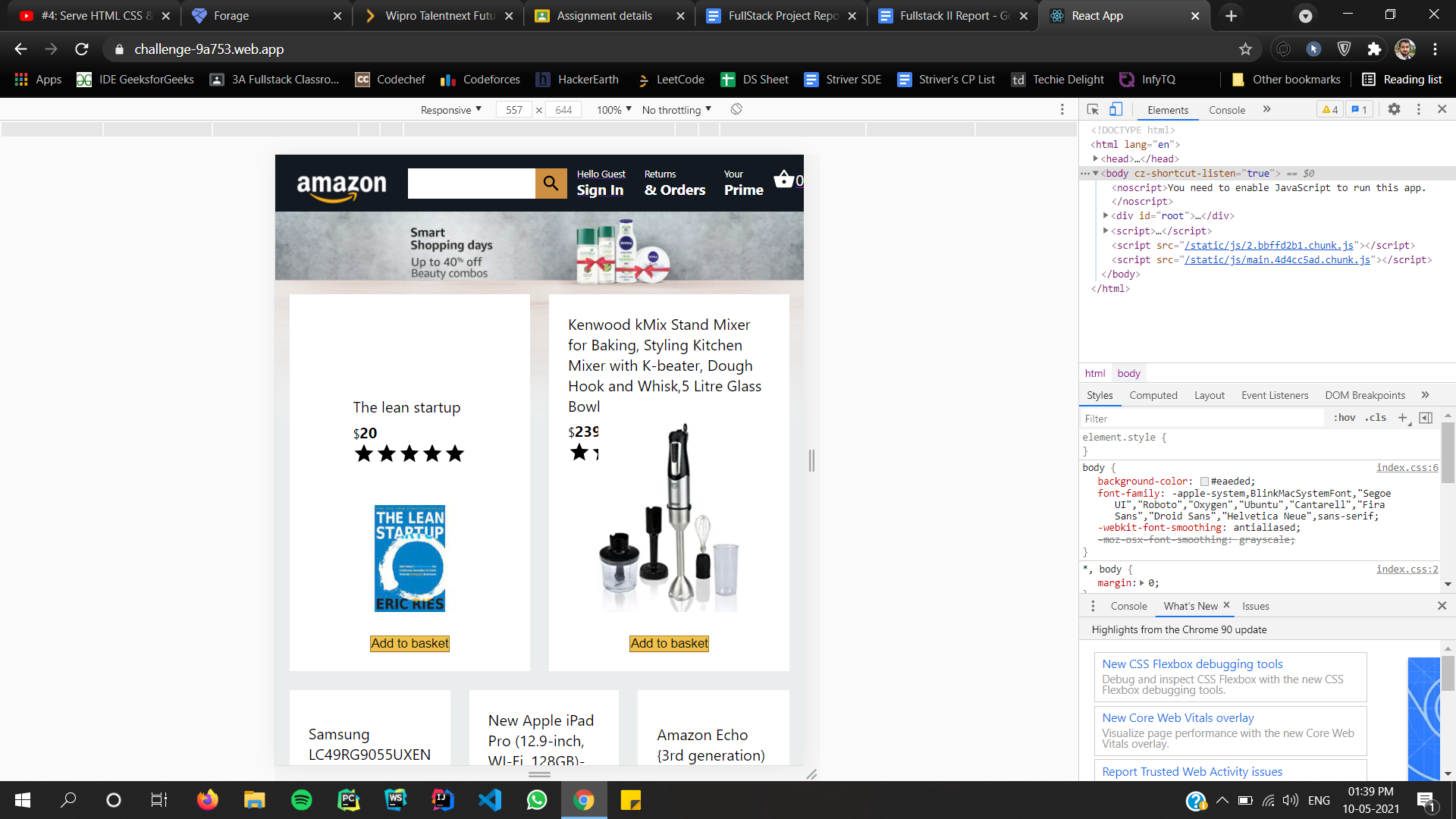
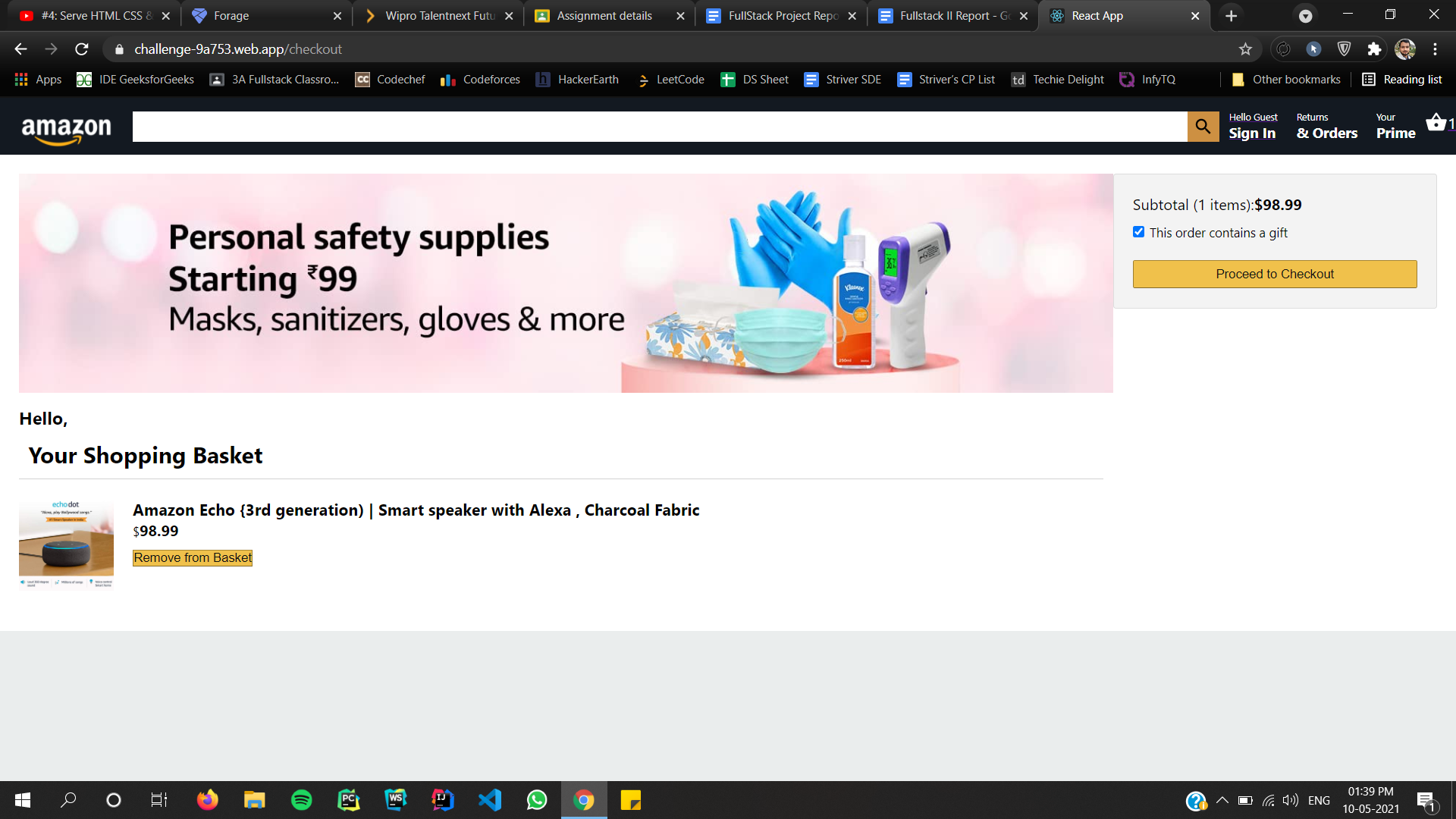
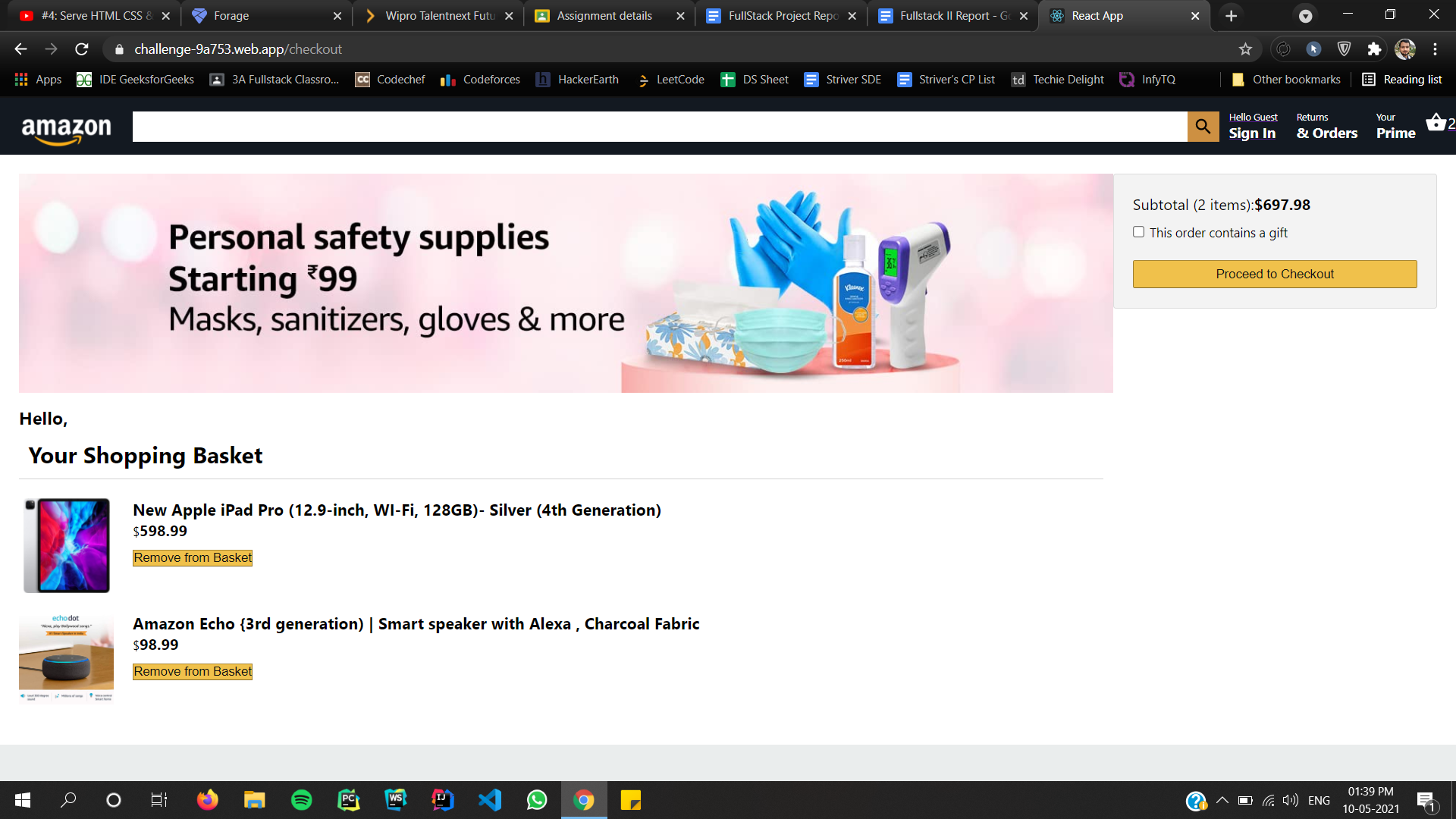
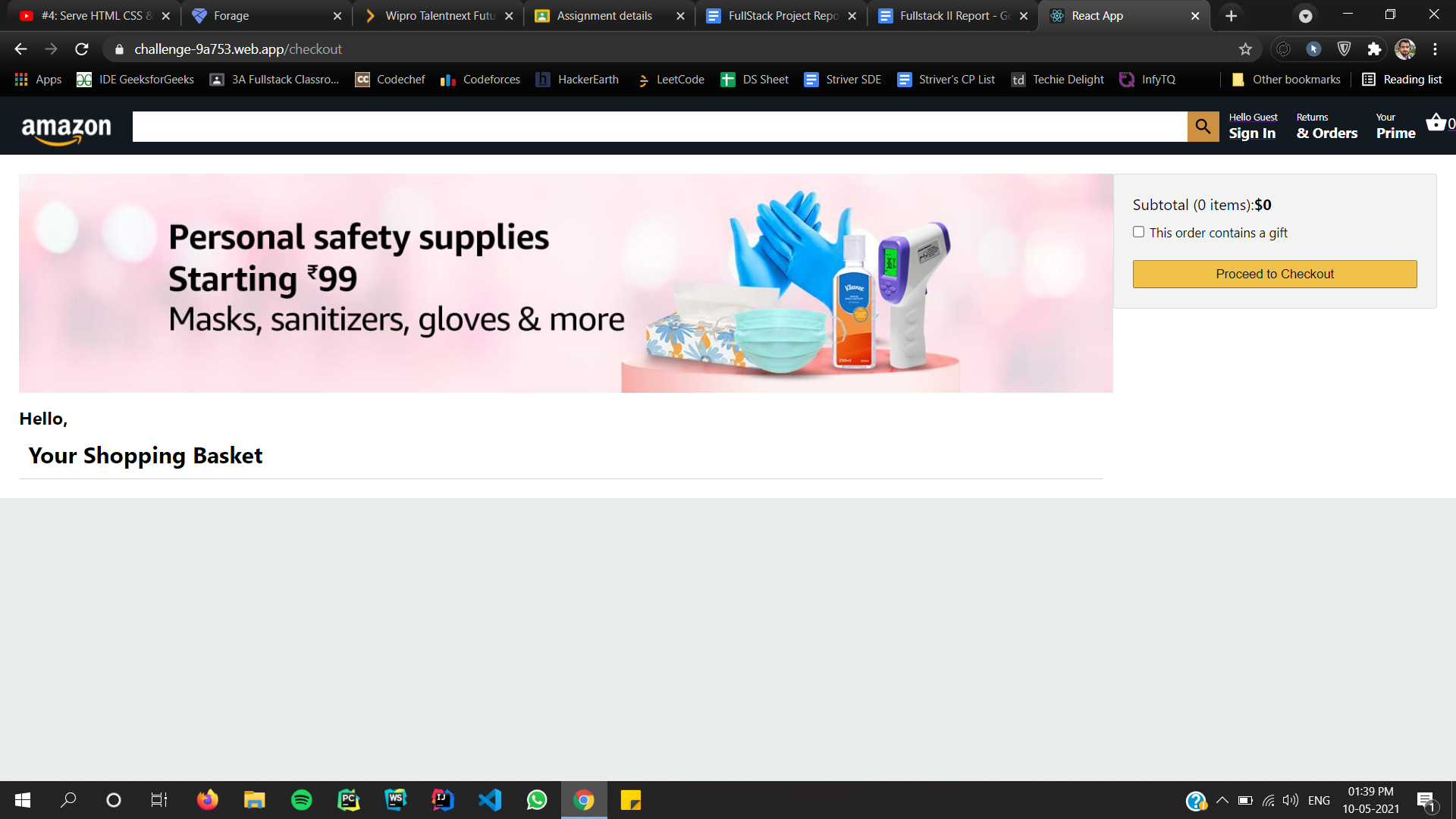
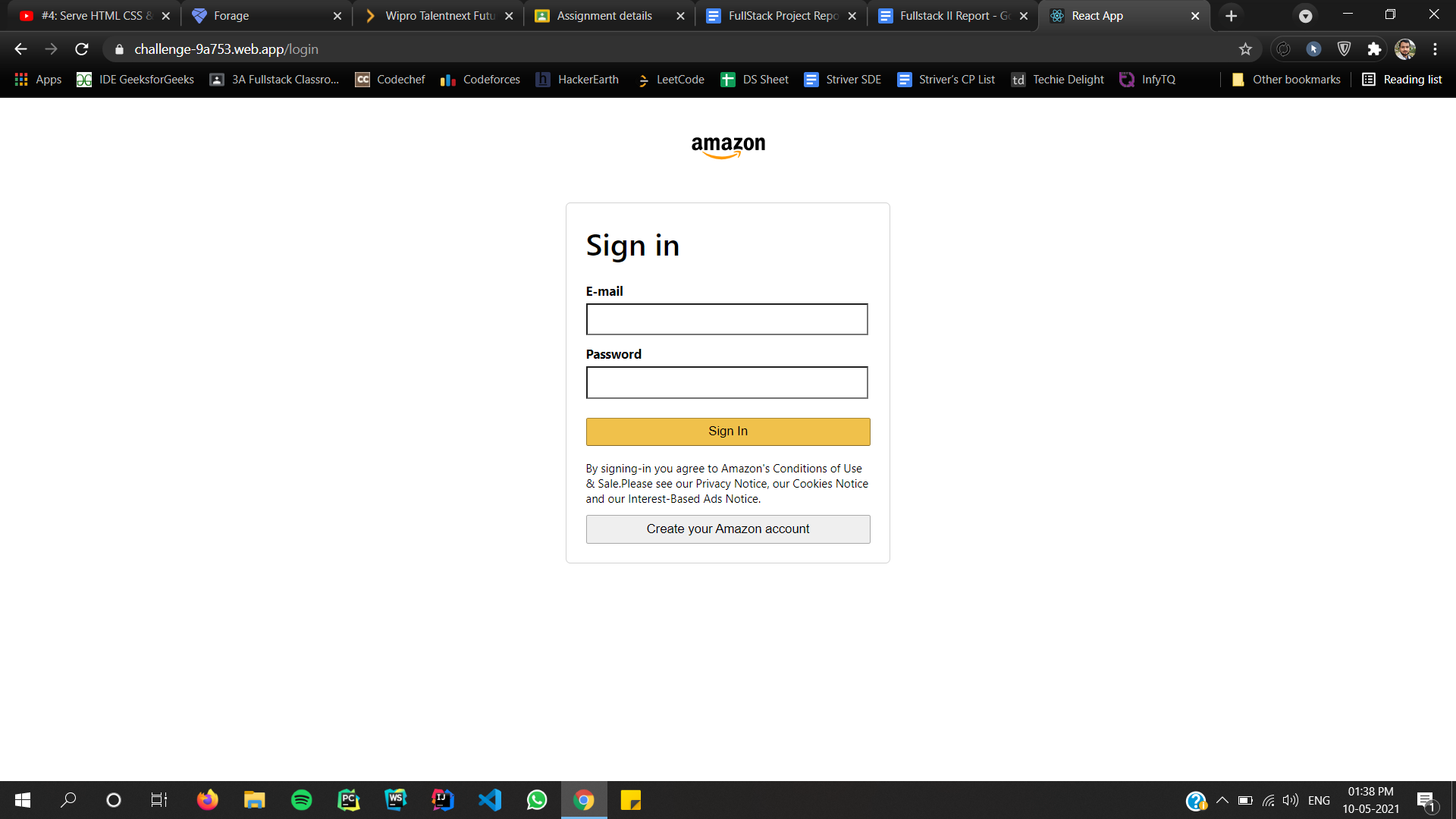
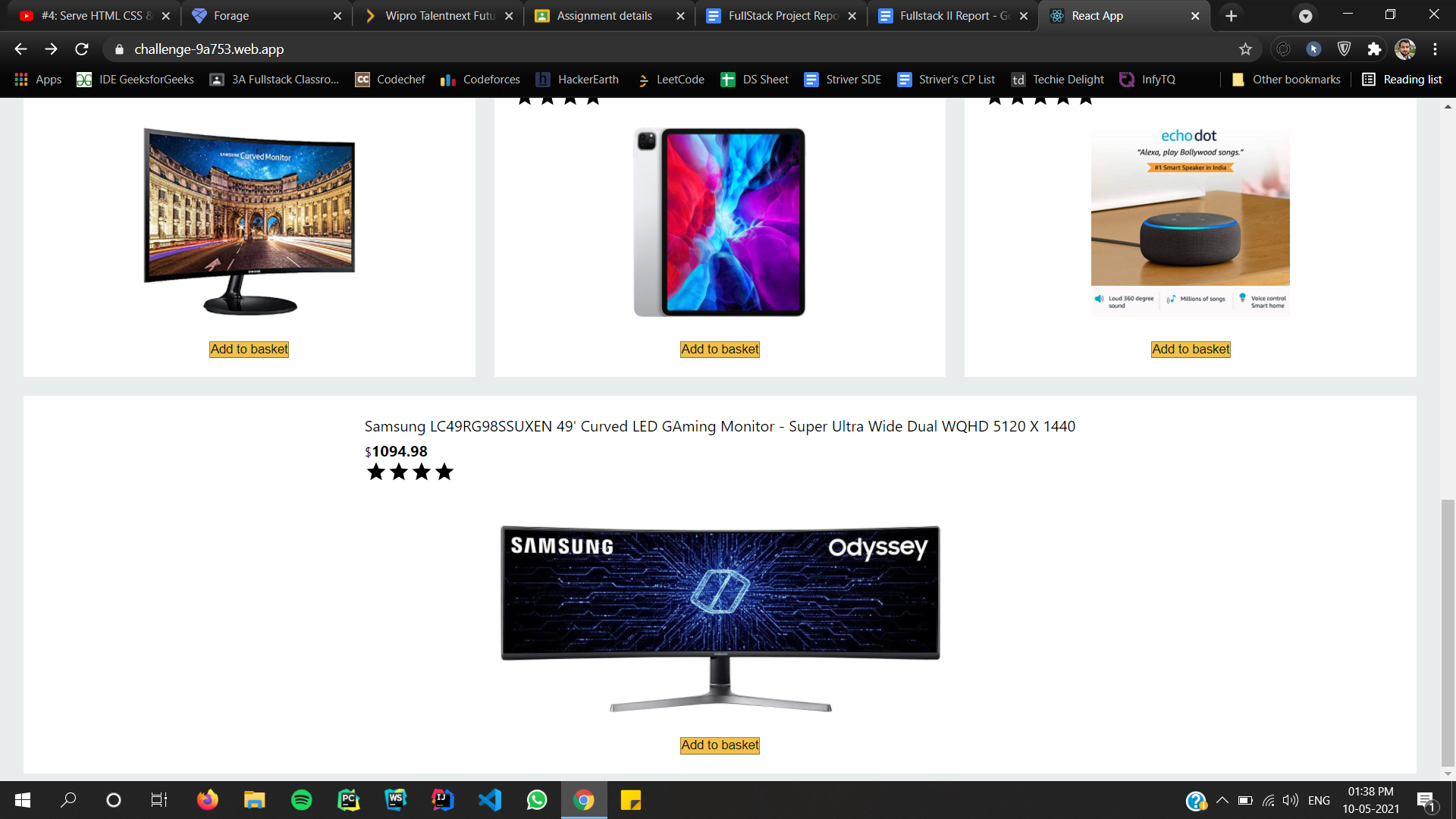
## **Software Quality Improvement:** The computer and the software are mainly used for complex and critical applications and a bug or fault in software causes severe losses. So a great consideration is required for checking for quality of software.

#### **Verification And Validation:** Verification means to test that we are building the product in the right way .i.e. are we using the correct procedure for the development of software so that it can meet the user requirements.Validation means to check whether we are building the right product or not.

**Chapter-5**

**Implementation and User Interface**





**Chapter-6**

**Reference & Bibliography**

* <https://www.youtube.com/>
* <https://www.javascript.com/>
* <https://animate.style/>
* <https://stackoverflow.com/>
* <https://www.reactjs.org>
* [https://getbootstrap.com/](https://getbootstrap.com/docs/4.0/getting-started/introduction/)
* <https://www.expressjs.com>

**Chapter-7**

**Links**

# 7.1 Project Github Link

https://github.com/Akansha0211/Amazon-Clone

# 7.2 Live Project Link

https://challenge-9a753.web.app/