MINI PROJECT – I

(2019-20)

Burger Shop Web Application

MID TERM REPORT

Institute of Engineering & Technology

Team Members

Ashi Saxena
(171500062)
Harshit Jaiswal
(171500126)

Supervised By

Mr. Pankaj Kapoor

Technical Trainer

Department of Computer Engineering & Applications

Mid Term Report

About the Project:

This project aims at creating a customized burger shop web app using React Js in which there will be a static website consists of about the shop, menu of burger with the prices and the gallery.
A page will be created in which customer can build a customized burger using ingredients like veggies , cheese and a variety of herbs and spices.
The activity goes the stages of ingredients selection, mixing, dividing and shaping. A detailed personalized step by step burger is produced so that customer can make their own burger.
After ordering , details will be deployed on firebase(database) where a shop owner can see all the details.

Motivation:

The main motivation of this project regards to making life easier of people and giving them more choice whenever you go to a burger joint you might like few ingredients of a burger and few not our project can help people select what they want to eat and not what the joint is selling that too in a cost effective way.

Future Prospects:

This Application helps customer to build their own burger with their own choices of ingredients and veggies easily.

Requirements:

a) **Hardware:**
- Latest configuration Laptop

b) **Software:**
- Visual Studio(Version 1.38)
- Brackets(Version 1.14)
- Notepad++ 7.7.1
- Firebase (Backend)

Github Id:

https://github.com/Harshitjais/burgershop

Methodology

Creating a Static Website

HTML5 is used to program the basic structure of the static website using different tags to fulfil different requirements.

HTML is a Hyper Text Markup Language which describes the structure of the webpage. It consists series of elements which tell the browser how to display the content. HTML elements are represented by tags. The browser does not display the tags, but use them to render the content of the page. HTML elements have attributes which provide additional information regarding elements. Attributes are always specified in the start tag.

HTML Tags

- All HTML documents must start with a document type declaration: <!DOCTYPE html>.
- The HTML document itself begins with <html> and ends with </html>.
- The <head> element is a container for metadata (data about data) and is placed between the <html> tag and the <body> tag.
- The <link> element is used to link to external style sheets.
- The <meta> element is used to specify which character set is used, page description, keywords, author, and other metadata.
- The visible part of the HTML document is between <body> and </body>.
- The HTML <p> element defines a **paragraph.**
- The `<div>` element is often used as a container for other HTML elements. It has no required attributes, but `style`, `class` and `id` are common.

HTML Attributes

- The link address is specified in the href attribute.
- The `id` attribute specifies a unique id for an HTML element. id value can be used by CSS and JavaScript to perform certain tasks for the element with the specific id value.
- class attribute is used to define equal styles for elements with the same class name.

Designing the Wesbite

The designing of the Static Website is programmed with CSS which is a language that describes the style of an HTML document. It describes how HTML documents should be displayed. CSS stands for Cascading Style Sheets. CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes. There are following three types of CSS**: -**

- Inline CSS.
- Internal CSS.
- External CSS.

Inline CSS: For Inline CSS every style content is in HTML elements. It is used for a limited section. Whenever the requirements are very small, inline CSS is used. It will affect only single elements. There is a lot of time consumed by that and it is not the best practice for a good programmer and the code will be quite large and very complex.

Internal CSS: In internal CSS the style of CSS is specified in the <head> section. This is internal CSS and it affects all the elements in the body section. Internal CSS is used in the condition when we want a style to be used in the complete HTML body. For that we can use style in the head tag.

External CSS: In External CSS, a .css file is created and used in the HTML page as per requirements. Generally external Cascading Style Sheets are used whenever the programmer has many HTML attributes. In external CSS, there is no need to rewrite the CSS style again and again in a complete body of HTML that inherits the property of the CSS file. The programmer simply has to create a .css file and he just have to link it to HTML page using <link> element and href attribute.

CSS Selectors: In CSS, selectors are patterns used to select the elements which are to be styled. Different selectors are as follows:

- **.class:** It selects all the elements within a specific class for styling.
- **#id**: It selects the element having specific id for styling.
- *: It selects all the elements for styling.
- **element:** It selects all the data within a specific element for styling.
- **:hover:** It selects links on mouse over for styling.
- **elements1 elements2:** It selects all elements2 within elements1 for styling

Adding a JavaScript

The working of the Static Website is programmed by JavaScript which is the programming language of HTML and Web. This language decides the behaviour of web pages.

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript is a lightweight, interpreted programming language that allows to build interactivity into otherwise static HTML pages.

React Js

A JavaScript library using for interface designing

React is a JavaScript library for building user interfaces. 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 or mobile applications, as it is optimal for fetching rapidly changing data that needs to be recorded.

Declarative

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.

Component-Based

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.

Updating to New Releases

Create React App is divided into two packages:

- create-react-app is a global command-line utility that you use to create new projects.
- react-scripts is a development dependency in the generated projects (including this one).

You almost never need to update create-react-app itself: it delegates all the setup to react-scripts.
When you run create-react-app, it always creates the project with the latest version of react-scripts so you'll get all the new features and improvements in newly created apps automatically.
To update an existing project to a new version of react-scripts, open the changelog, find the version you're currently on (check package.json in this folder if you're not sure), and apply the migration instructions for the newer versions.
In most cases bumping the react-scripts version in package.json and running npm install in this folder should be enough, but it's good to consult the changelog for potential breaking changes.
We commit to keeping the breaking changes minimal so you can upgrade react-scripts painlessly.

Folder Structure

After creation, your project should look like this:

```
my-app/
README.md
node_modules/
package.json
public/
  index.html
  favicon.ico
src/
  App.css
  App.js
  App.test.js
  index.css
  index.js
  logo.svg
```

For the project to build, **these files must exist with exact filenames**:

- public/index.html is the page template;
- src/index.js is the JavaScript entry point.

You can delete or rename the other files.

You may create subdirectories inside src. For faster rebuilds, only files inside src are processed by Webpack.
You need to **put any JS and CSS files inside src**, otherwise Webpack won't see them.
Only files inside public can be used from public/index.html.
Read instructions below for using assets from JavaScript and HTML.
You can, however, create more top-level directories.
They will not be included in the production build so you can use them for things like documentation.

Available Scripts

In the project directory, you can run:

npm start

Runs the app in the development mode.
Open http://localhost:3000 to view it in the browser.

The page will reload if you make edits.
You will also see any lint errors in the console.

npm test

Launches the test runner in the interactive watch mode.
See the section about running tests for more information.

npm run build

Builds the app for production to the build folder.
It correctly bundles React in production mode and optimizes the build for the best performance.
The build is minified and the filenames include the hashes.
Your app is ready to be deployed!

See the section about deployment for more information.

npm run eject

Note: this is a one-way operation. Once you eject, you can't go back!
If you aren't satisfied with the build tool and configuration choices, you can eject at any time. This command will remove the single build dependency from your project.
Instead, it will copy all the configuration files and the transitive dependencies (Webpack, Babel, ESLint, etc) right into your project so you have full control over them. All of the commands except eject will still work, but they will point to the copied scripts so you can tweak them. At this point you're on your own.
You don't have to ever use eject. The curated feature set is suitable for small and middle deployments, and you shouldn't feel obligated to use this feature. However we understand that this tool wouldn't be useful if you couldn't customize it when you are ready for it.

Backend

Firebase is a **Backend**-as-a-Service (BaaS) app development platform that provides hosted **backend** services such as a realtime database, cloud storage, authentication, crash reporting, machine learning, remote configuration, and hosting for your static files.

Build apps fast, without managing infrastructure

Firebase gives you functionality like analytics, databases, messaging and crash reporting so you can move quickly and focus on your users.

Implementation Details

The implementation is divided in two parts:

1. Backend Development:

FireBase:

Firebase is a **Backend**-as-a-Service (BaaS) app development platform that provides hosted **backend** services such as a realtime database, cloud storage, authentication, crash reporting, machine learning, remote configuration, and hosting for your static files.

Build apps fast, without managing infrastructure

Firebase gives you functionality like analytics, databases, messaging and crash reporting so you can move quickly and focus on your users.

2. Frontend Development:

The technology used for developing frontend is HTML, CSS, JAVASCRIPT,REACT JS. React is a JavaScript library for building user interfaces. 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 or mobile applications, as it is optimal for fetching rapidly changing data that needs to be recorded.

Contribution Summary

We have divided the project into three parts :

- Backend Development
- Frontend Development
- Documentation

1. Harshit Jaiswal is responsible for Frontend and Backend.
2. Ashi Saxena is responsible for Frontend and Documentation.

Progress Till Date and the Remaining Work

Till date we have completed front end static Website and started learning about ReactJs and Firebase.

Remaining Part:

- ReactJs FrontEnd
- Database Connections
- Validation
- Testing

Screenshots

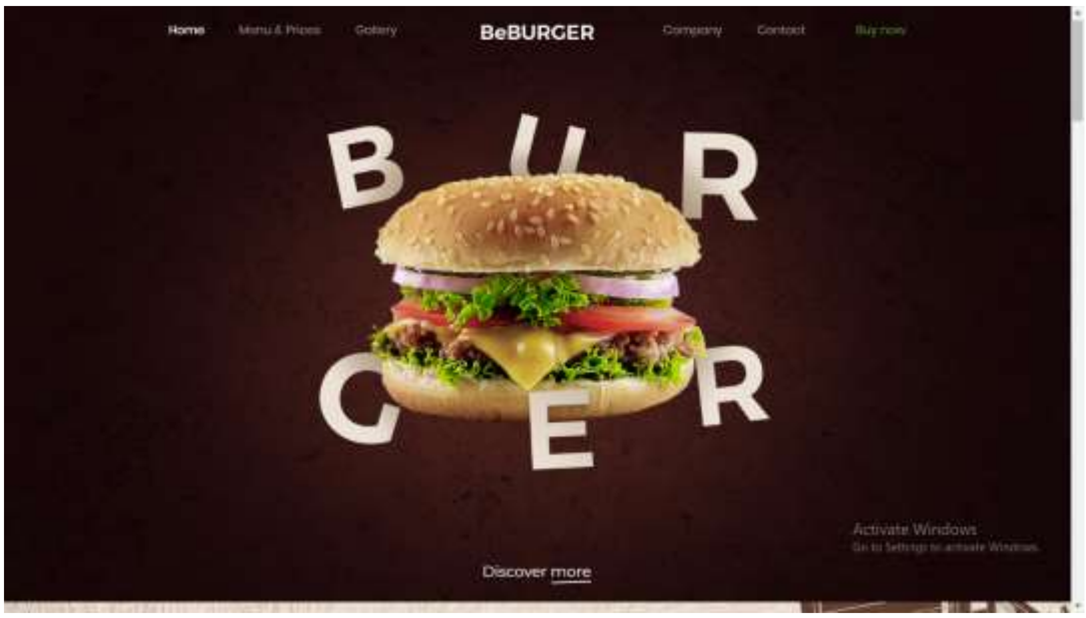


Fig 1. Home Page

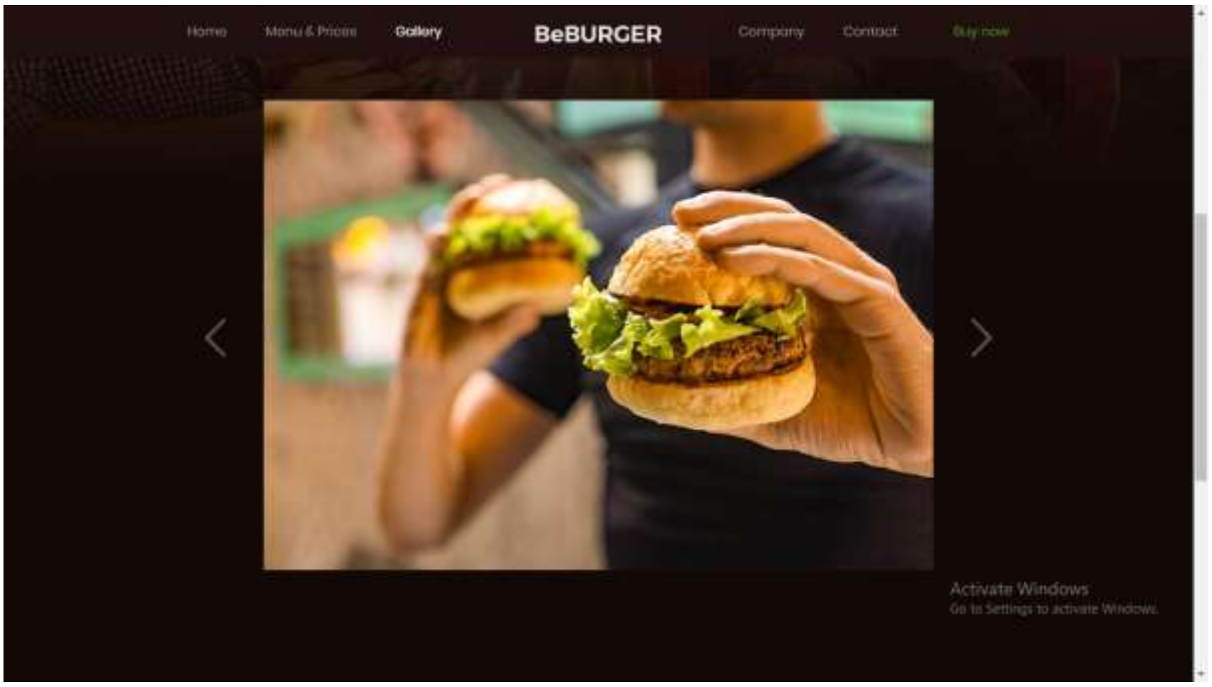
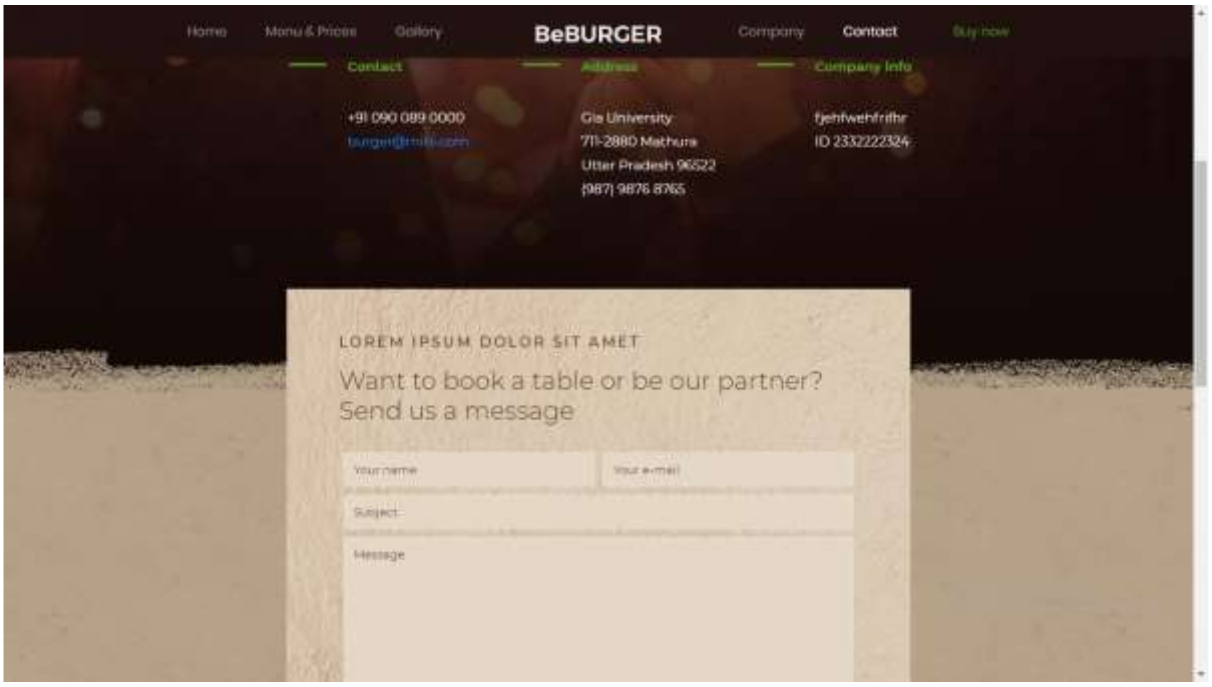


Fig 2. Gallery



Contact

+91 090 089 0000
burger@mail.com

Address

Gla University
711-2880 Mathura
Utter Pradesh 96522
(987) 9876 8765

Company info

fjehfwehfrifhr
ID 2332222324

LOREM IPSUM DOLOR SIT AMET

Want to book a table or be our partner?
Send us a message

Fig 3. Contact

References

- https://firebase.google.com/
- https://reactjs.org/
- https://github.com/facebook/create-react-app
- https://www.w3schools.com/