

**DEVELOP A RESPONSIVE USER INTERFACE WITH
REACT
A NAAN MUDHALVAN REPORT
SUBMITTED BY
MEENA R 912421106009
MOHAMED ABDULLA I 912421106010
BACHELOR OF ENGINEERING
IN
ELECTRONICS AND COMMUNICATION ENGINEERING**



**SHANMUGANATHAN ENGINEERING COLLEGE
ARASAMPATTI, PUDUKKOTTAI – 622 507**

**YEAR & SEMESTER : IV & VII
SUBJECT CODE :NM1050
COURSE NAME :SaaS (SOFTWARE AS A SERVICE)**



**ANNA UNIVERSITY::CHENNAI 600 025
NOV/DEC 2024**

BONAFIDE CERTIFICATE

Certified that this Naan Mudhalvan report of **“DEVELOP A RESPONSIVE USER INTERFACE WITH REACT”** is the bonafide work of **“MEENA R (912421106009)” & “MOHAMED ABDULLA I (912421106010)”** who carried out the project work under my guidance.

SIGNATURE

Mrs. D. LATHA M.E.,
NAAN MUDHALVAN COORDINATOR
ASSISTANT PROFESSOR,
Department of Electronics &
Communication Engineering,
Shanmuganathan Engineering College,
Arasampatti – 622 507

SIGNATURE

Dr. A.MUTHU MANICKAM M.E., Ph.D.,
HEAD OF THE DEPARTMENT
ASSISTANT PROFESSOR,
Department of Electronics &
Communication Engineering,
Shanmuganathan Engineering College,
Arasampatti – 622 507

Submitted for the Internship viva-voice on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGMENT

At this pleasing moment having successfully completed our internship report, we wish to convey our sincere thanks to our beloved chairperson **Mrs. PICHAPPA VALLIAMMAL**, correspondent **Dr. P. MANIKANDAN B.E**, director(Academic) Shri **M.SHANMUGANATHAN**, director(Administration) Shri **M. PICHAPPA** and honourable secretary **Mr. M. VISWANATHAN** for their extensive support.

I thankful to our principal **Dr. KL. MUTHURAMU M.E(W.R)., M.E(S.E)., Ph.D., FIE., M.I.S.T.E.**, Shanmuganathan engineering college, for providing the opportunity to conduct our project.

I extend our gratitude to **Dr. A.MUTHU MANICKAM M.E., Ph.D.**, the head of the department and our internship co-ordinator of Electronics and communication engineering for providing a valuable suggestion and supports given through the study.

Gratitude never fails. We are grateful to our dynamic and effective internal guide **Asst. Prof. Mrs. D. LATHA, AP/ECE, M.E.**, for his valuable innovative suggestion, constructive interactions, constant encouragement and valuable helps that have been provided us throughout the project.

I also express my heartfelt thanks to all other staff members of Electronics communication engineering Department for their support. Above all, we thank our parents, for affording us the valuable education till now.

ABSTRACT

Responsive web design is an approach to web design that makes web pages render properly on a variety of devices and screen sizes. In other words, responsive design ensures that your web page looks great and works well no matter what device your users are viewing it on. Flexible layouts a responsive layout is one that can adapt to any screen size. This means that your web page will look good on both large and small screens, and you won't need to create separate versions for each. Media queries are a CSS feature that allows you to specify different styles for different devices. This means that you can use different CSS rules for different devices, making your page more responsive. Flexible images and media another important aspect of responsive design is using flexible images and media. This means using assets that can scale to fit any screen size, and using media that can be played on both small and large screens. When it comes to typography, it's important to make sure that your fonts are legible and look good on all devices. This can be achieved by using responsive typography, which is a technique that allows you to use different font sizes for different devices. Navigation is one of the most important aspects of a website, and it needs to be easy to use on all devices. When creating a responsive website, it's important to make sure that your navigation is clear and easy to use. Responsive design is an important part of creating a website that works well on all devices. By using a responsive layout, media queries, and flexible images and media, you can create a website that looks great and is easy to use no matter what device your users are using.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
1.	INTRODUCTION TO RESPONSIVE USER INTERFACE	
	1.1 INTRODUCTION TO RESPONSIVE UI	1
	1.2 PRINCIPLES OF RESPONSIVE UI	1
	1.3 IMPORTANT OF RESPONSIVE UI	2
2.	HOW BULID A RESPONSIVE USER INTERFACE	
	2.1 BUILD A RESPONSIVE UI	3
	2.2 FRONT-END DEVELOPMENT	3
3.	PROJECT IMPLEMENTATION	
	3.1 HTML CODES FOR PROJECT IMPLEMENTATION	7
	3.2 CSS CODES FOR PROJECT IMPLEMENTATION	8
	3.3 REACT FOR LOGIN PAGE	9
	3.4 OUTPUT FOR THE RESPONSIVE USER INETFACE	11
4.	CONCLUSION	
	4.1 CONCLUSION	12

CHAPTER 1

INTRODUCTION TO RESPONSIVE USER INTERFACE

Introduction to Responsive User Interface (UI):

A **Responsive User Interface (UI)** is an approach to web design and development that ensures a seamless and optimized user experience across a variety of devices, screen sizes, and orientations. The goal of responsive UI design is to create flexible layouts that automatically adapt to the device's screen, providing users with the best possible experience regardless of whether they're browsing on a smartphone, tablet, laptop, or desktop.

In today's digital age, where users access websites and applications on a wide range of devices, responsive design has become essential. It improves usability, accessibility, and performance by ensuring that a site or app looks good, functions well, and is easy to navigate no matter what device a user is using.

Principles of Responsive UI

1. Fluid Layouts

A fluid layout uses percentage-based widths instead of fixed pixel widths. This allows the content to adjust based on the screen size. For example, a 50% width on a desktop will look different on a smaller screen but will still fill the available space proportionally.

2. Flexible Media

Images, videos, and other media should be scalable and flexible. By using CSS to define media properties (like `max-width: 100%`), images and videos resize automatically to fit the screen size. This ensures that the content doesn't break or overflow from its container on smaller screens.

3. Media Queries

Media queries are a fundamental tool for responsive UI design. They allow designers to apply different CSS rules based on the characteristics of the device, such as its width, height, resolution, and orientation. With media queries, you can adjust layout styles, font sizes, and other elements for optimal viewing on each device.

4. Mobile-First Design

Mobile-first design is a strategy where web pages are initially designed for mobile devices, and then the layout and features are progressively enhanced for larger screen sizes. This ensures that the core content and functionality are easily accessible on mobile devices, which often have more constraints (such as smaller screen sizes and slower internet speeds).

5. Responsive Navigation

Navigation elements, like menus and buttons, must also be flexible and user-

friendly. On mobile devices, this often means simplifying the menu into a "hamburger" style or collapsible menu, while on larger screens, the menu can be displayed in a horizontal layout. The idea is to make navigation easy to use regardless of the screen size.

6. Typography and Readability

Typography plays a key role in a responsive UI. Text should be legible on all devices, and font sizes should scale proportionally to the screen size. CSS techniques like relative units (em, rem, vw) allow font sizes to be flexible and adapt to different screen resolutions.

Important of Responsive UI

- **Improved User Experience (UX):** A responsive design ensures that users have a positive experience no matter what device they use. It eliminates the need for users to zoom, scroll horizontally, or struggle with broken layouts on mobile devices.
- **Cost-Effectiveness:** Instead of designing separate websites for different devices (desktop, mobile, tablet), a single responsive website or app adapts to all device types. This reduces development, maintenance, and update costs.
- **SEO Benefits:** Search engines like Google prioritize mobile-friendly websites in their rankings. Having a responsive site means better visibility and ranking in search results, especially as mobile usage continues to grow.
- **Faster Load Times:** By optimizing content for different screen sizes and resolutions, responsive UI design can help reduce unnecessary data transfer. This is especially important for mobile users with slower internet connections.

CHAPTER 2

How to Build a Responsive UI

Build a Responsive UI:

In this project of developing an Responsive UI the major part is taken from the **Front-end** . Here are the tools that I used to develop a **Responsive UI**.

Front-end Development:

- HTML (HYPER TEXT MARKUP LANGUAGE)
- CSS (CASCADING STYLE SHEET)
- REACT JS

FRONT-END DEVELOPMENT:

HTML:

The full form of HTML is Hyper Text Markup Language.

HTML is the standard markup language used to create and design web pages. It provides the basic structure for web content, such as text, images, links, and other media. HTML uses a system of tags (or elements) to define the different parts of a webpage, making it possible for browsers to render content properly.



Key Features of HTML:

1. **Structure of Web Pages:** HTML defines the structure and layout of a webpage, using elements like headings, paragraphs, links, lists, tables, images, and forms.
2. **Markup Language:** As a markup language, HTML doesn't perform any computation; it merely describes the structure of a document using tags.
3. **Tags:** HTML uses a series of tags enclosed in angle brackets (e.g., <div>, <h1>, <p>,) to format text, create hyperlinks, embed media, and more.
4. **Semantic Elements:** HTML provides semantic tags like <header>, <footer>, <article>, and <section>, which help structure content in a meaningful way, both for browsers and search engines.
5. **HTML5:** The latest version of HTML, HTML5, includes new elements (like <video>, <audio>, and <canvas>) and features for better multimedia integration, mobile optimization, and performance improvements.

CSS:

CSS (Cascading Style Sheets) is a stylesheet language used to control the layout and presentation of HTML elements on a webpage. It separates the content (HTML) from the design (CSS), allowing for better organization, flexibility, and maintenance of the website's visual presentation.



USES OF CSS:

1. **Separation of Content and Style:** CSS allows you to define the presentation (fonts, colors, margins, etc.) separately from the HTML content, making the code cleaner and easier to maintain.
2. **Consistency:** With CSS, you can apply the same style across multiple pages, ensuring consistency across your website.
3. **Responsive Design:** CSS enables you to create responsive layouts that adapt to various screen sizes (mobile, tablet, desktop).

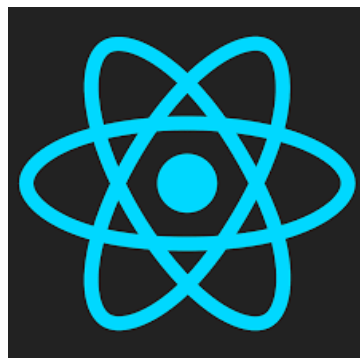
4. **Customization:** CSS offers a wide range of styling options to customize the look and feel of your website, including fonts, colors, animations, transitions, and positioning.

Types of CSS:

1. **Inline CSS:** Applied directly within an HTML element using the style attribute.
2. **Internal CSS:** Defined within a <style> tag in the <head> section of an HTML document.
3. **External CSS:** Linked to an external CSS file using the <link> tag. This is the most common method, especially for larger websites.

REACT JS:

ReactJS is an open-source JavaScript library created by Facebook for building dynamic and interactive user interfaces. Unlike traditional JavaScript frameworks, ReactJS takes a component-based approach to UI development.



Key aspects of ReactJS:

- React allows you to build encapsulated components that manage their state and render DOM elements. This makes it easy to break complex UIs into smaller, reusable pieces.
- It uses a virtual DOM that updates efficiently behind the scenes when data changes. This provides major performance benefits compared to manipulating the real DOM directly.
- React follows a unidirectional data flow paradigm. Data flows in one direction down through nested components via props. This makes the logic easier to understand.
- JSX is an optional syntax extension that allows writing HTML-like code directly within JavaScript. This gives you the full power of JavaScript within markup.

Benefits of using ReactJS:

- Increased developer productivity — React's component architecture and modern API result in less code to achieve the same outcomes.
- Improved maintenance — Component isolation and one-way data flow lead to UIs that are more predictable and easier to debug.
- Great performance — The virtual DOM provides huge rendering speed improvements by minimizing costly DOM operations.
- Reusable components — Encapsulated components with well-defined interfaces can be shared across projects.
- Universally applicable — React can be used for simple view layers but also complex enterprise applications. It scales up well.

Responsive Design Principles:

Responsive web design is an approach to designing websites that allows them to adapt and change based on different screen sizes. As more users access the web on a variety of devices like desktops, tablets, and phones, responsive design has become essential for modern web development. To achieve this, there are some core responsive design principles:

Fluid Layouts:

Using relative width units (percentages or em units) instead of fixed pixel widths for layout elements allows them to shrink or expand based on the viewport width. This creates a flexible, fluid layout that can adapt as the screen size changes.

Flexible Images:

Making images fluid, so they can scale and resize rather than having a fixed width and height. This prevents images from spilling outside their containers or looking pixilated on different screen sizes.

Responsive Navigation:

The navigation menu needs to adapt and possibly change to a collapsed mobile menu on smaller screens. Using a flexible or off-canvas navigation pattern is key.

Responsive Data Visualizations:

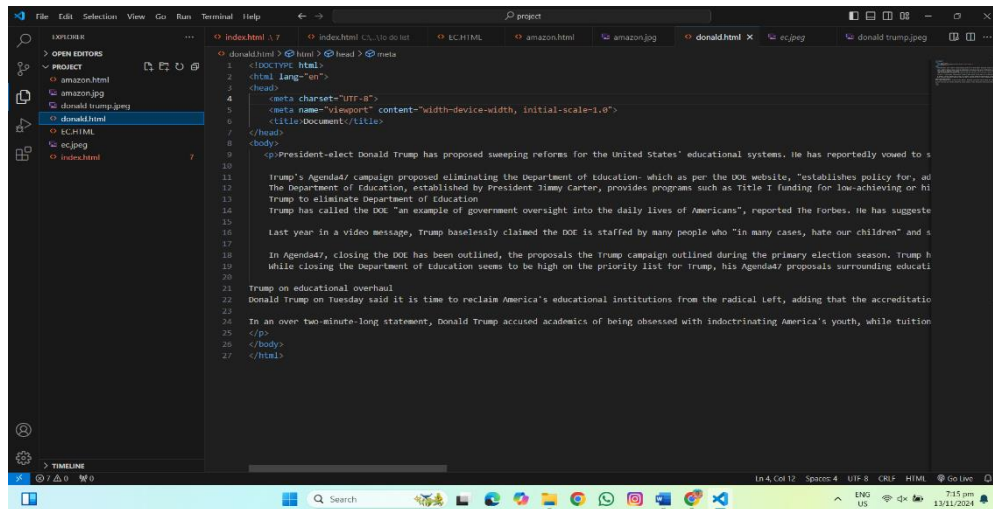
Charts, graphs, and maps may need to change their layout for smaller screens. The information should adapt to the available screen space.

HTML CODE FOR PROJECT IMPLEMENTATION:



Syntax

12



Responsive Images:

Responsive images play a key role in responsive websites. These are images that can adjust their size, getting bigger or smaller, based on the width of the browser. By being responsive, images enhance user experience across different devices with varying screen sizes. The following are the techniques to use the responsive images.

Using width Property:

The image can be responsive & scale up & down with the help of CSS width property by setting its value as 100%.

Syntax

```

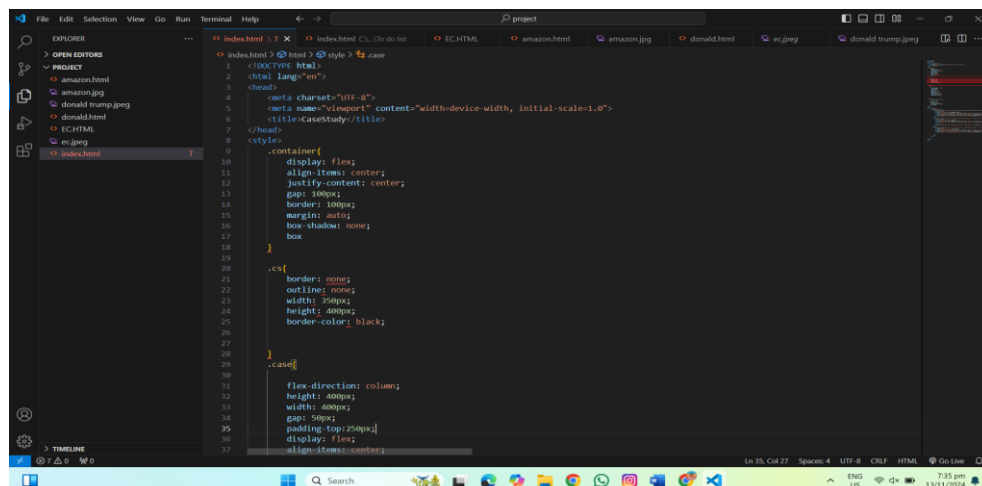
```

CSS CODES FOR PROJECT IMPLEMENTATION:

- By taking the time to learn about responsive design, you can ensure that your website looks great no matter what device it's being viewed on.
- All you need to do is make sure that your CSS is properly formatted and that your ReactJS code is written in a way that makes sense for responsive design. Formatting your CSS correctly is crucial for responsive design.
- You need to make sure that your code is written in a way that will allow your website to resize and respond properly to different screen sizes. A good way to do this is to use the "reactive" design principle.
- This principle states that your code should be written in a way that makes it easy to change and adapt the design of your website in response to different screen sizes. This way, you can easily

change the way your website looks without having to rewrite your entire codebase.

- CSS has been around for a long time and is well-supported by modern browsers. It allows you to create media queries that specify different styles for different screen sizes. Even if you're new to web development, you can quickly get up to speed with the basics of CSS media queries and start creating responsive designs.



- This code will place item1 in the first and second columns of the first row, item2 in the third column of the first row, and item3 in the first column of the second row.
- CSS subgrid allow you to create a grid within a grid. This is useful when you have a nested layout, where the child elements of a grid container should have a different grid structure than the parent container.
- To create a nested grid system, you can use the '**subgrid**' keyword in the **grid-template-columns** and **grid-template-rows** properties of the child element.

REACT FOR LOGIN PAGE :

- React Responsive is a library that makes it easy to apply different styles to your React components based on the screen size. This is useful for creating responsive designs that automatically adjust to the size of the screen on which they are being viewed.
- The library provides ways to wrap your existing React components and apply the appropriate styles based on the screen size. This can help you avoid having to manually write rules in your CSS, and makes it easy to create responsive designs without a lot of extra effort.

- There are several advantages to using the React-Responsive library when developing websites with ReactJS:

- **Simplified code:** React-Responsive makes it easy to apply responsive styles to your components without having to write complex rules in your CSS. This can make your code more readable and easier to maintain.
- **Consistency:** By using the React-Responsive library, you can ensure that your responsive designs are consistent across all screen sizes. This can help you avoid having to write separate code for different screen sizes, which can save time and reduce the risk of errors.
- **Reusability:** The components provided by the React-Responsive library can be easily reused in different parts of your website, which can help you avoid having to duplicate code and make your designs more modular and flexible.
- **Community support:** React-Responsive is an established library that is widely used by the ReactJS community. This means that you can easily find help and support if you have any questions or encounter any problems when using the library.

- Overall, using react-responsive is a great way to create responsive React apps. It makes the process of creating responsive layouts much easier and provides a great way to customize the look of your app for different screen sizes.

- To use the react-responsive library, we first need to install it using npm.

```
import App from "../Test_2.svelte";
```

```
const app = new App({  
  target: document.body,  
  props: {  
    name: "world",  
  },  
});
```

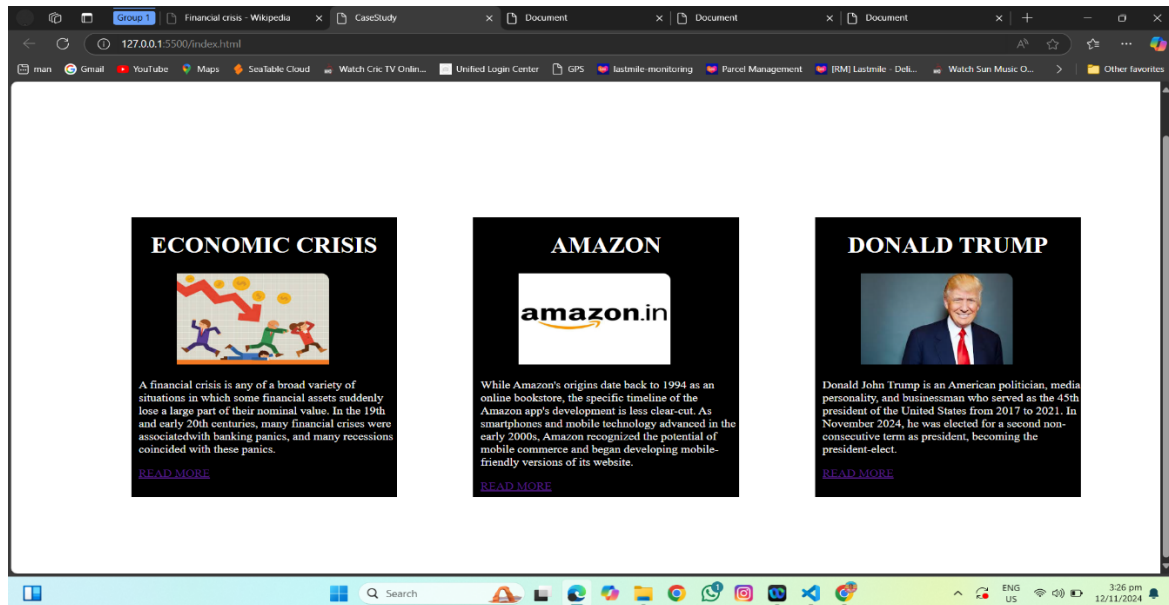
```
export default app;
```

```
if (import.meta.hot) {  
  import.meta.hot.accept();  
  import.meta.hot.dispose(() => {  
    app.$destroy();  
  });  
}
```

- Check the website's layout and design is one of the most important aspects of a responsive website is its layout and design.
- When testing the responsiveness of a website, make sure to check how the layout and design adapt to different screen size

OUTPUT:

The Output that we get in the project implementation is the figures ,codes and the HTML ,CSS and relevant Front-end that should be obtain from Visual Studio code.



CHAPTER 4

CONCLUSION

CONCLUSION:

➤ In conclusion, React's component-based architecture, coupled with its efficient virtual DOM, makes it an ideal tool for building responsive user interfaces. By strategically utilizing CSS media queries, flexbox, and grid layouts, developers can create web applications that seamlessly adapt to various screen sizes and orientations. Component-Based Approach encapsulating UI elements into reusable components promotes modularity and simplifies the process of managing layout changes across different screen sizes.

➤ Virtual DOM React's virtual DOM optimizes rendering performance, ensuring smooth transitions and reducing load times, especially on devices with limited resources. Rich Ecosystem is a vast array of libraries and tools, such as React Responsive, Material-UI, and Bootstrap, provide pre-built components and utilities to accelerate responsive development. Community Support is a large and active community offers valuable resources, tutorials, and support, enabling developers to learn and troubleshoot effectively.

While React provides a robust foundation for building responsive UIs, it's crucial to consider these additional aspects for an optimal user experience. ARIA Attributes employ ARIA attributes to provide additional context for screen readers and other assistive devices. Keyboard Navigation ensure that users can navigate and interact with your application using only a keyboard. Testing and Debugging responsive Testing use tools like BrowserStack or LambdaTest to test your application across various devices and screen sizes. Debugging Tools utilize browser developer tools to inspect and debug layout issues. Unit Testing write unit tests to verify the behavior of your components under different screen sizes.