DS_ARENA

A Thesis Submitted In Partial Fulfillment of the Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

BY

Shivam Nerwal

(1900290140033)

and

Shubham Kumar Vishnoi

(1900290140036)

Under the supervision of

MR. Naresh Chandra

(Assistant Professor)

Department of Computer Applications

KIET Group Of Institutions

Submitted to

Department of Computer Applications

KIET Group of Institutions

Ghaziabad

DECLARATION

I hereby declare that the work presented in this report entitled "**DS_Arena**", wascarried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. I have given duecredit to the original authors/sources for all the words, ideas, diagrams, graphics,computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources. I affirm that no portion of my work is plagiarized, and the programs and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, we shall be fully responsible and answerable.

Shivam Nerwal(1900290140033)
Shubham Kumar Vishnoi(1900290140036)

ACKNOWLEDGEMENT

At the outset, we would like to thank our guide and advisor**, Mr. Naresh Chandra Assistant Professor (Department of Computer Applications),** for giving usanopportunity towork on this challenging topic and providing us ample and valuable guidance throughout the Project.

Without his encouragement and constant guidance,we would not have able to finish this project. He has been always a source of inspiration and motivator for innovative ideas duringthe entirespan of this work.

Wearegratful**Dr.AjayKumarShrivastava,ProfessorandHead,DepartmentofComputerApplications,KIETGroupofInstitutions,Ghaziabad**forprovidingallthenecessaryresources tocarryoutthisProjectwork.

We will be failing in our duty if we don't acknowledge the people behind this work to give usmoral and psychological support. Our special thanks to our parents for their endless care and constant support.

ShivamNerwal(1900290140033)

ShubhamKumarVishnoi(1900290140036)

CERTIFICATE

Certified that Name of student **Shivam Nerwal (1900290140033), Shubham Kumar Vishnoi (1900290140036)**hascarried out the project work presented in this

Report entitled **"DS_ARENA"** for Master of Computer Applications from Dr. APJ Abdul Kalam Technical University, Lucknow under my supervision. The project embodies results of original work, and studies are carried out by the student himself and the contents of the project do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Shivam Nerwal (1900290140033)

Shubham Kumar Vishnoi(1900290140036)

From the best of my knowledge this is certify that information provided by candidate is correct.

Mr.Naresh Chandra
Assistant Professor
Department of Computer Applications
KIET Group of Institutions

Signature of Internal Examiner **Signature of Extermal Examiner**

ABSTRACT

DS_ARENA is a single pae web application which provides Information about basic data structures this appliaction is based on React which is a java script framework.
The main idea behid this application is to learn a new technology like to know more about single page applictaions along with revising data structures concepts.

To provide audience a creadible source where they can brush up the basic concept of Data structures. A simple language, interactive images and gif along with an impressive user interface is used to make our application more attractive and user friendly in future we are planning to add a feedback section in our application so that we can improve it according to users feedback. In future we introduce backend in our application for that we will choose Node Js.

Contents

Title Page	1
Declaration	2
Acknowledgement	3
Certificate	4
Abstract	5
Contents	6
List of Figures	7

Chapter 1. Introduction

1.1. Description
1.2. Scope
1.3. Future Scope
1.4. Identification of need
1.5. Software and Technology used
1.6. Project schedule and planning

Chapter 2. Feasiblity Study

2.1. Introduction

2.2. Technical Feasiblity
2.3. Economical Feasiblity
2.4. Leagal Feasiblity
2.5. Operational Feasiblity
2.6. Benefits of Conducting Feasiblity Test

Chapter 3. Planing and WorkFlow

3.1 Planning of project
3.2 SDLC (Software Development Life Cycle)
3.2 Description
3.3 Workflow

Chapter 4. Coding

4.1 Public / Index.html
4.2 Src/Components/Stack/StackData
4.3 Src/Data
4.4 Src/NavBar
4.5 AllCont.jsx
4.6 App.jsx
4.7 index.js

Chapter 5. Testing

5.1 Introduction
1. objectives
2. Principle

5.2 Levels
1. Unit Testing
2. Integration Testing
3. System Testing

Chapter6. Future Scope

Chapter7. Conclusion and Bibliography

7.1 Conclusion

7.2 Bibliography

7.3 Github Link of Project Repository

List Of Figures

Figure 1. VS Code 11

Figure 2. React 12

Figure 3. JSON 12

Figure 4. JSX 13

Figure 5. WaterFall Model 20

Figure 6. Front Page 21

Figure 7. Front Page 2 22

Figure 8. Content Page 22

Figure9. Testing Levels 53

Chapter 1 : INTRODUCTION

1.1 Description

Data structures is one of the most important part in the journey of the student of computer applications and computer science student lots of people have a will to learn but they are not able to get a proper source where they can embark on and get all the credible content under a single source.

Our application provides the description of each basic data structure along with the programs that how to create a data structure from scrach and the code of their basic oparations In future we are planning to include a section where users can contribute if they want to improve any content and after approval their contribition can be added in the application.

We mainly use React in our application and learn alot from this that how to use React

hooks and how to utilise them to make a single page application. We use some interactive images and GIFs along with a impressive user interface which is created by using css.

For storing content we are using json and whole the data is stored in the json from if we want to make changes then just by changing that json file we can change that content.

1.2 Scope

The main objective of the project is to develop a single page web application which will provide a credible information to a user and if a user want the program of any data structure like the self -refrential structure to create a data structure from scrach then he can easily get that information. And the application is based on react so instead of fetching the whole page the only component is fetched and desired data is rendered on the screen of user.

1.3 Future Scope

- In future we want to include backend in our application for this we are planning to use Node js that's why we included it in our technical stack.
- Login secton for users we are planning to include a login section for users.
- Feedback feature such that if any user want a imporovement in any content then he just submit the content and after the approval his changes are visible in the main content.
- We also want to include a discussion section so that the users who are visiting the application can discuss in that section and a good sharing of quality knowledge take place.

1.4 Identification of Need

We want to learn a new technolgy which is used in industry so that in future when we learn it we would not face any difficulty so we choose React and for future scope when we include backend in for that we choose Node JS. The below points shows the requirements of data structures so we choose this topic of our web application so that alon
.

- Data structure is one of the fundamental and essential part in the field of computer science if any person wants to develop good software then the knowledge of data structures plays a important role.
- In code optimization and to write write effective approach of code the knowledge of complexities and data structure plays a very pivotal role.
- When any computer science student approaches towards placements then Data Structures along with competetive coding problems is the most important round to crack

These all points inclined us more towards this topic and we choose it for our project. We are focused and in future to we want to improve the quality as well as content of our application we are consistantly working towards in that direction and learning new things from it.

1.5 Software and Technology Used

1. Visual Studio Code(VS Code)

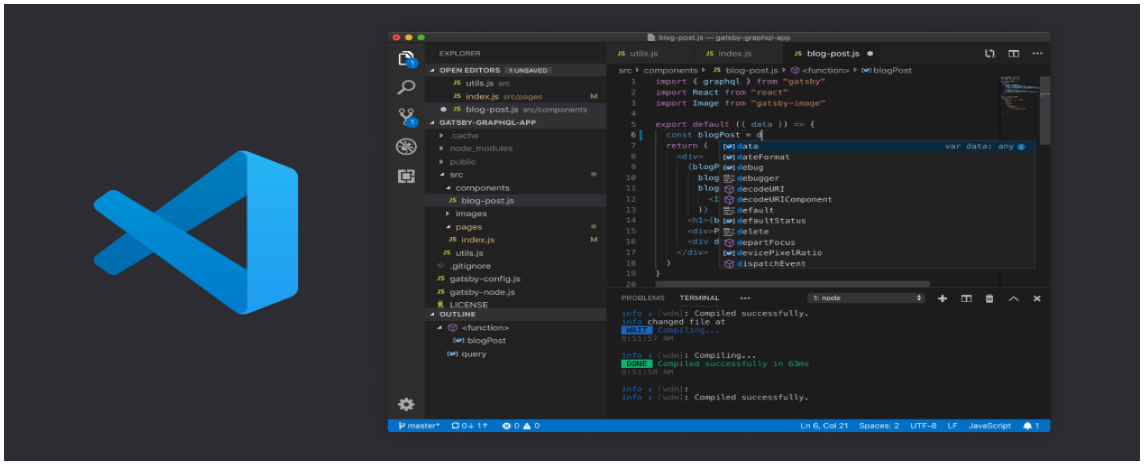

Figure 1.VS Code

Visual Studio Code (VS code) is a streamlined code editor developed by microsofts with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows. It is widely used in Industry and runs smoothly on macOS, Linux, and Windows.

2. React JS

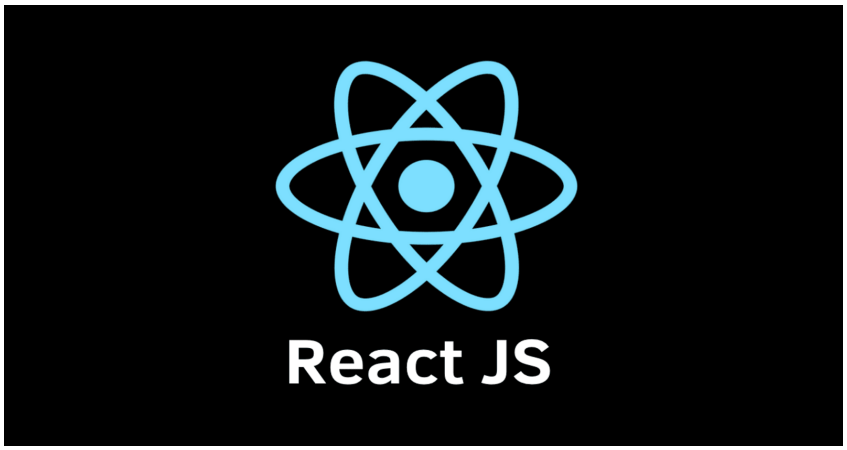

Figure 2. React

React is a free and 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 or mobile applications.

2.1. JSON (Java Script Object Notation)

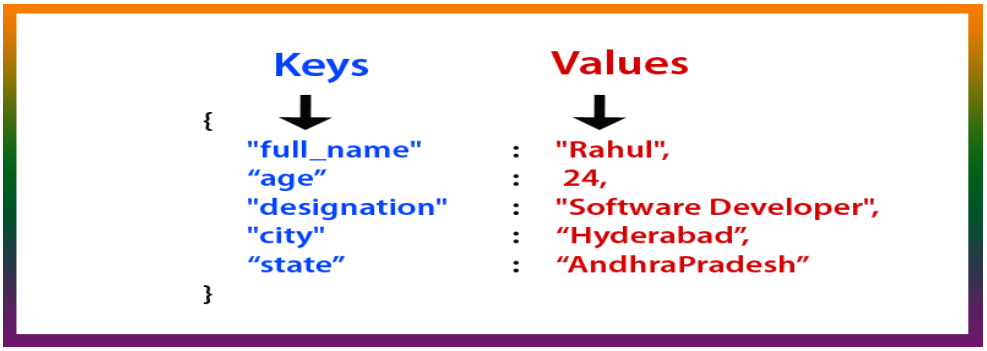

Figure 3. json

JSON (Java Script Object Notation) is an open standard file format and data inter-change format that uses human-readable text to store and transmit data objects consist-ing of attribute–value pairs and arrays. In our project we use JSON to store data so that if any user wants to change thenn it can be easily changed from json file without chang-ing the whole program.

2.2. JSX (JavaScript XML)

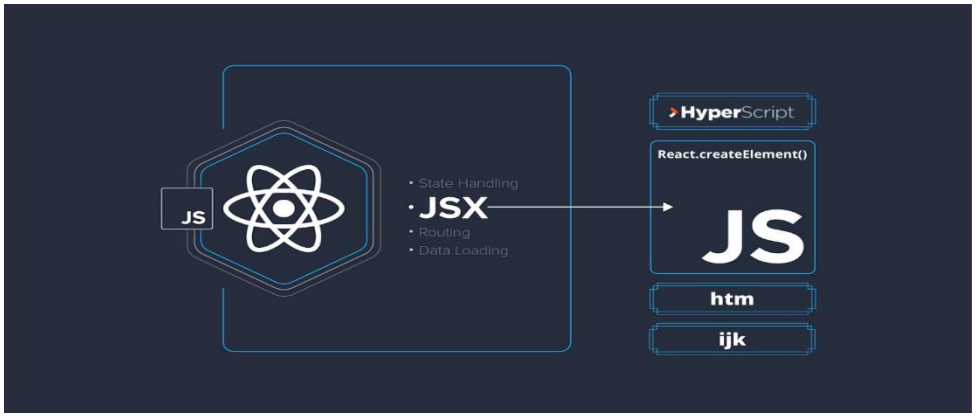

Figure 4. jsx

JSX(Java Script XML), and it is a syntax extension to JavaScript.It is recommend usedit with React to describe what the UI should look like. JSX may remind you of a template language, but it comes with the full powe

r of JavaScript.JSX producesReact“elements”.

1.6. Project Schedule and Planning

The objective of software project planning is to provide a framework that enables the manager tomake reasonable estimates of resources, costs and schedule. These estimates are made within alimited time frame at the beginning of a software project and should be updated regularly as theproject progresses. In addition, estimates should attempts to define “best case” and “worst case”scenariossothat projectoutcomes can bebounded.

The first activity in software project planning is the determination of software scope.Function and performance allocated to software during system engineering should be assessed toestablish a project scope that is ambiguous and understandable at management and technical levels.Softwarescopedescribesfunction,performance, constraints,interfacesandreliability.

During early stages of project planning, a microscopic schedule is developed. This type ofschedule identifies all major software engineering activities and the product functions to which theyare applied. As the project gets under way, each entry on the macroscopic schedule is refined intodetailedschedule.Herespecificsoftwaretasks are identified andscheduled.

Schedulinghasfollowingprinciples:

1. Compartmentalization:theprojectmustbecompartmentalizedintoanumberofmanageableactivities and tasks.
2. Interdependency:theinterdependenciesofeachcompartmentalizedactivityortasksmustbedetermined.
3. Timeallocation: each taskto bescheduled mustbeallocated some numberofwork

units.

4. Effortvalidation:every projecthas adefined numberof staffmembers.

5. Definedresponsibili-
ties:everytaskthatisscheduledshouldbeassignedtoaspecificteammember.

6. Definedoutcomes:everytaskthatisscheduled shouldhaveadefinedoutcome.

CHAPTER 2. Feasiblity Study

2.1. Introduction

Feasibility of the system in an important aspect, which is to be considered. The system needs to satisfy the law of economic, which states that the maximum output should be yielded in minimum a vailable resources.

A feasibility analysis evaluates the project's potential for success; therefore, perceived objectivityis an essential factor in the credibility of the study for potential investors and lending institutions.There are five types of feasibility study separate areas that a feasibility study examines, described below.

2.2. Technical Feasibility

Thisassessmentfocusesonthetechnicalresourcesavailabletotheorganization.Ithelpsorganizations determine whether the technical resources meet capacity and whether thetechnicalteam iscapable ofconverting the ideas intoworking systems.Asan exaggerated example, anorganization wouldn't want to try to put Star Trek's transporters in their building currently, thisprojectisnottechnicallyfeasible.

2.3 Economic Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizationsdetermine the viability, cost, and benefits associated with a project before financial resources areallocated. It also serves as an independent project assessment and enhances project credibility—helpingdecision-makersdeterminethepositiveeconomicbenefitstotheorganizationthattheproposedproject will provide.

2.4 LegalFeasibility

This assessment investigates whether anyaspectof the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let's say an organizationwants to construct a new office building

in a specific location. A feasibility study might reveal theorganization'sideal location isn'tzonedforthat typeof busines. That organization has just saved considerable time and effort by learning that their project was notfeasiblerightfrom the beginning.

2.5 OperationalFeasibility

This assessment involves undertaking a study to analyze and determine whether and how well theorganization'sneedscanbemetbycompletingtheproject.Operationalfeasibilitystudies also examine how a project plan satisfies the requirements identified in therequirementsanalysisphaseof systemdevelopment.

SchedulingFeasibility

This assessment is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.

When these areas have all been examined, the feasibility analysis helps identify anyconstraintsthe proposedproject may face,including:

a. InternalProjectConstraints:Technical,Technology,Budget,Resource,etc.
b. InternalCorporateConstraints:Financial,Marketing,Export,etc.
c. ExternalConstraints:Logistics,Environment,Laws,andRegulations,etc.

2.6. Benefitsofconductingafeasibility study

i. Improvesprojectteams' focus
ii. Identifiesnewopportunities

iii. Providesvaluableinformationfora“go/no-go”decision

iv. Narrowsthebusinessalternatives

v. Identifiesavalid reasontoundertaketheproject

vi. Enhancesthesuccessratebyevaluatingmultipleparameters

vii. Aidsdecision-makingontheproject

viii. Identifiesreasonsnottoproceed

Chapter 3. Planning and workflow

3.1 Planning

The objective of software project planning is to provide a framework that enables the manager tomake reasonable estimates of resources, costs and schedule. These estimates are made within alimited time frame at the beginning of a software project and should be updated regularly as theproject progresses. In addition, estimates should attempts to define “best case” and “worst case”scenariossothat projectoutcomes can bebounded.

The first activity in software project planning is the determination of software scope.Function and performance allocated to software during system engineering should be assessed toestablish a project scope that is ambiguous and understandable at management and technical levels.Softwarescopedescribesfunction,performance, constraints,interfacesandreliability.

During early stages of project planning, a microscopic schedule is developed. This type ofschedule identifies all major software engineering activities and the product functions to which theyare applied. As the project gets under way, each entry on the macroscopic schedule is refined intodetailedschedule.Herespecificsoftwaretasks are identified andscheduled.

Schedulinghasfollowingprinciples:

1. Compartmentalization:theprojectmustbecompartmentalizedintoanumberofmanageableactivitiesand tasks.

2. Interdependency:theinterdependenciesofeachcompartmentalizedactivityortasksmustbedetermined.

3. Timeallocation: each taskto bescheduled mustbeallocated some numberofwork units.

4. Effortvalidation:every projecthas adefined numberof staffmembers.

5. Definedresponsibilities:everytaskthatisscheduledshouldbeassignedtoaspecificteammember.

6. Definedoutcomes:everytaskthatisscheduled shouldhaveadefinedoutcome.

3.2 SDLC (Software Developmet Life Cycle)

ThewaterfallmodelwasselectedastheSDLCmodelduetothefollowingreasons-

1. Requirementswereverywelldocumented,clearandfixed
2. Technology was adequately understood
3. Simple and easy to understand and use.
4. Therewerenoambiguousrequirements
5. Easytomanageduetotherigidityofthemodel.Eachphasehasspecificdeliverablesanda review process.
6. Clearlydefinedstages
7. Wellunderstoodmilestones.Easytoarrangetasks

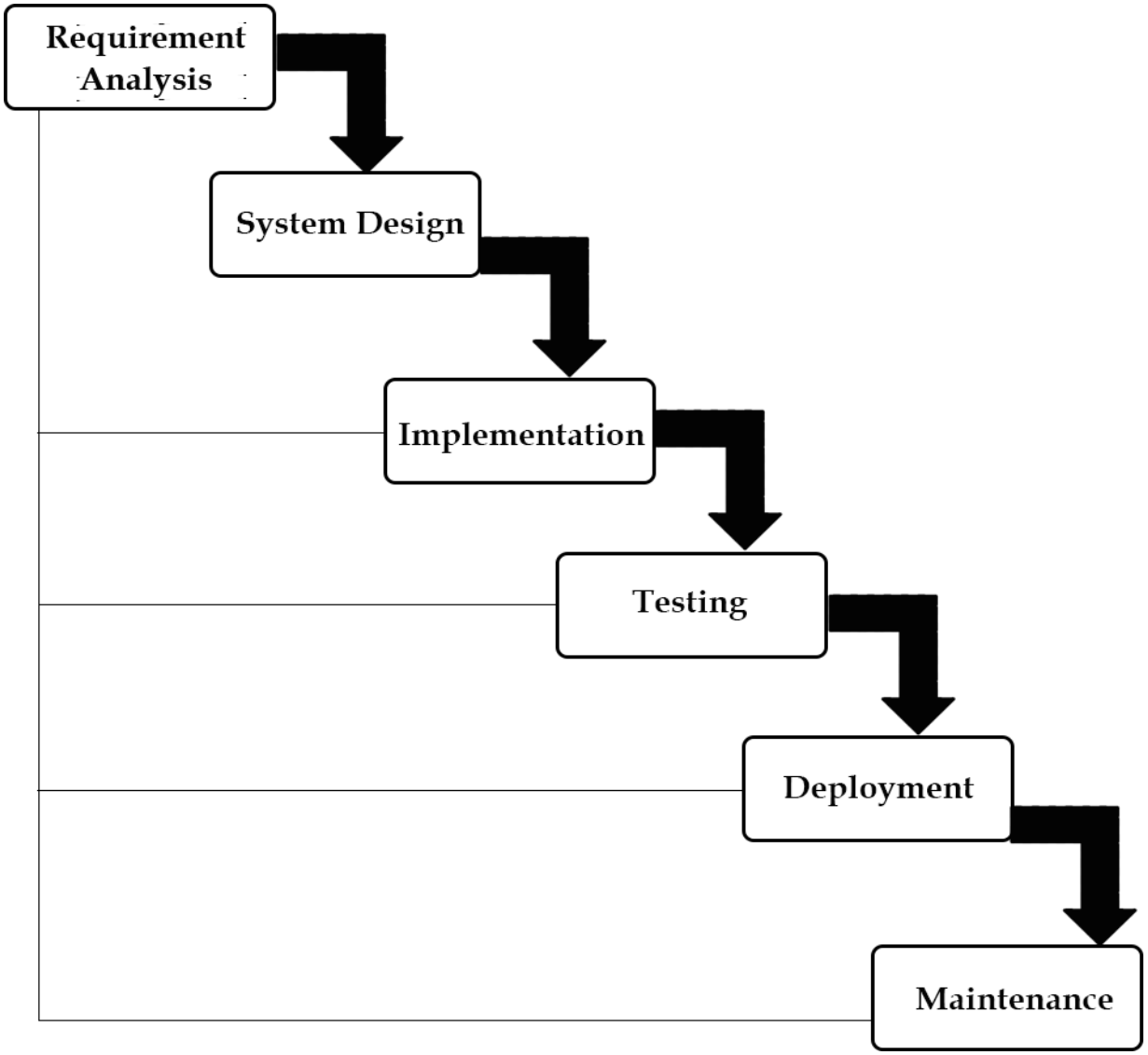

Figure 5 waterfall Model

3.3 Workflow

AS the react is a component oriented language we divided our project into different - different components this give us a better planning and a proper methodology to write program this way to break the code into components help us to understand that how react only cahnges and render a single component instead of whole page when we apply a proper coding technique in our program.

The first page of our application concicts a header from which we will go where we want (some feature are left for future scope)

And on the first page each data structure portion is shown by a beautiful user interface of card based layout as shown in figure below-

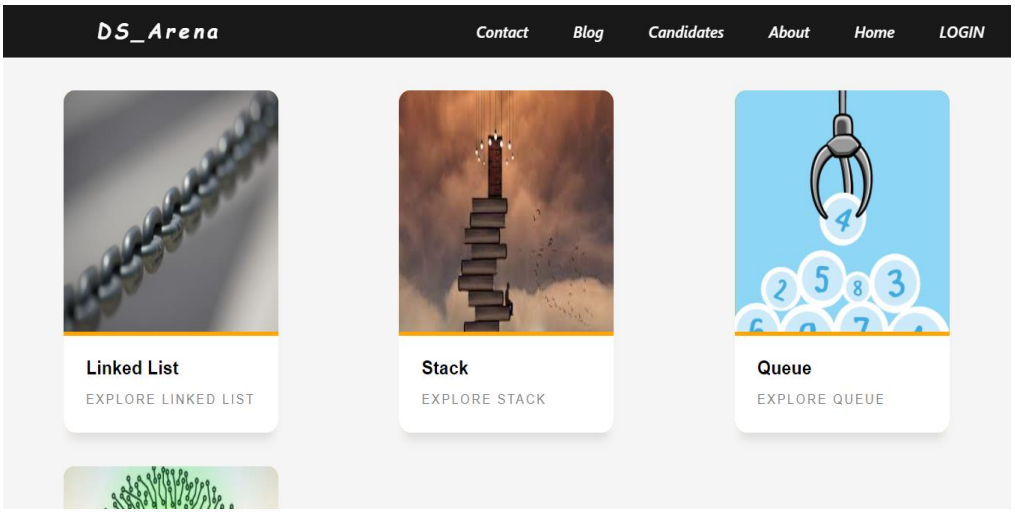

Figure 6. Front Page

If we hover on any card it get transformed and give a interactive and attractive look and feel as shown in figure below -

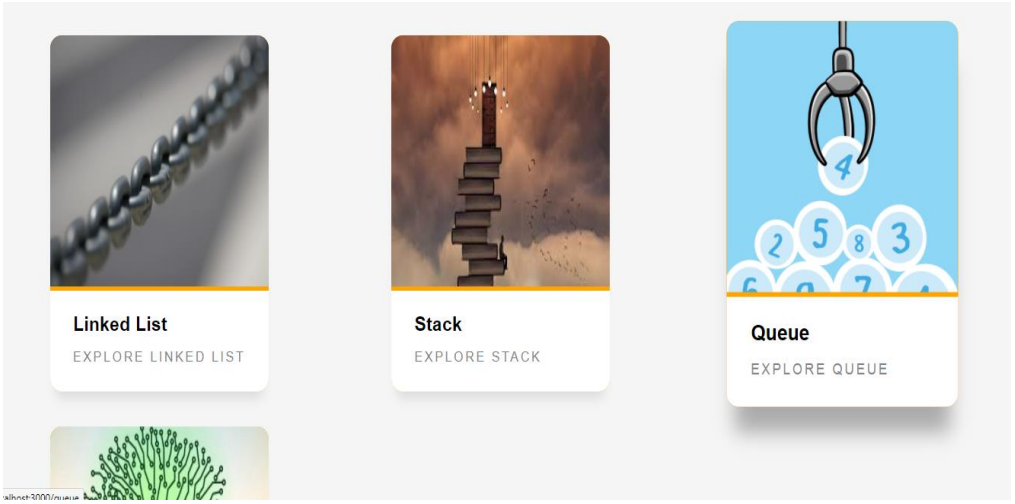

Figure 7 Front Page 2

After clicking on that particular card the user will enter into the section of that data structure as shown in figure below–

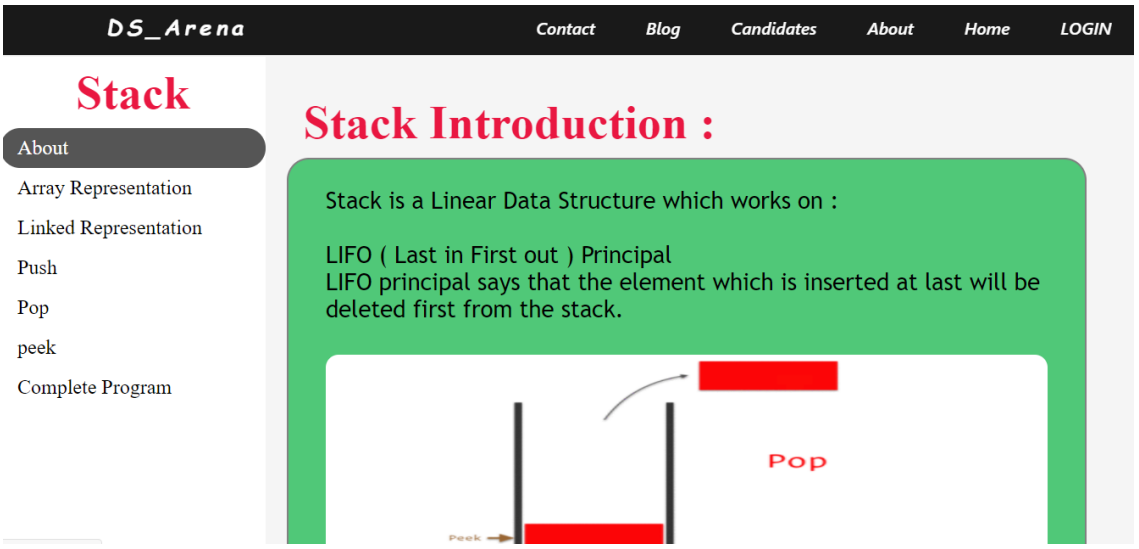

Figure 8 Content Page

As you click on any item in the navbar you will not see the browser refreshing and this shows our single page application is working smoothly.

In future we are planning and working to add backend in our application for this we are selected Node Js and in future login form and discussion box will be introduced in our project.

Chapter 4. Coding

As we show above design and how we finalise the working this all could be possible by the help of coding or programming and it is not incorrect to say that coding is the back-bone of any small or large project.
As react is a component oriented langauge we divided our project in various **functional components** and then we coded them to achieve our goal.

The designing is done by **css3** and **React Hooks** are used to make more functionality possible to achieve.
Almost whole the textual data like definations, image address, gifsare stored in **JSON** format and the react script is coded in **JSX.**

4.1. Public/ index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      us-
er's mobile device or desktop. See https://developers.google.com/web/fundamentals/we
b-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the build.
      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
    <title>React App</title>
  </head>
```

```
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
    <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.
      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run `npm start` or `yarn start`.
      To create a production bundle, use `npm run build` or `yarn build`.
    -->
  </body>
</html>
```

4.2. Src/Components/Stack/StackData

1. ArrayRepresentation

```
import Nav from "./StackNavBar";
import "./Stack.css";
import Data from "./StackData/Data.json";
import Img1 from "./Stack_Images/Array_Representation.png" ;

function NewlineText(props) {
```

```
  const text = props.text;
  return text.split('\n').map(str => <p>{str}</p>);
 }

const ArrayRepresentation = () => {
      const elem = <>
      <Nav />
      <br/>
      <h2 className="StackContainer" >
       <NewlineText text = {Data[3].content} />

      </h2>
      <p className="About">
      {Data[3].data}
      <br/>
      <br/>
      <NewlineText text = { Data[3].data2 } />
      <br/>
      <br/>
      <NewlineText text = {Data[3].data4}/>
      <br/>
      <NewlineText text = {Data[3].data3}/>
      <br/>

      <img src={ Img1 } className="Cont_Img" alt="Array_Representstion"/>
      <br/>
      <br/>
      <NewlineText text = {Data[3].content2}/>
      <br/>
      <NewlineText text = {Data[3].data5}/>
      </p>
      </>
      return elem ;
```

```
}
export default ArrayRepresentation ;
```

2. LinkedRepresentation

```
import Nav from "./StackNavBar";
import Img from "./Stack_Images/Linked_Representation.png";
import "./Stack.css";
import Data from "./StackData/Data.json";
const LinkedRepresentation = () => {

      const elem = <>
      <Nav />
      <h2 className="StackContainer">
      {Data[4].content}
      </h2>
      <p className="About">
      <img src={Img} className="Cont_Img" alt="Linked_Representation"/>
      </p>

      const Footer = ()

   </>
   return elem;
}
export default LinkedRepresentation;
```

3. LinkedRepresentation (CSS)

```
body{
   box-sizing: border-box;
```

```
}
.imgclass{
    margin: 23px;
    padding: auto;
    margin: -89px;
    margin-top: -420px;
    padding: 0px;
    margin-left: 26px;

}
.arimg{

    display: block;
    margin-left: auto;
    margin-right: auto;
    width: 50%;
    box-sizing: border-box ;
    border-radius: 23px;
    border: 4px solid black;
    cursor: pointer;
    opacity: 0.8;
    float: left center;
    margin: 466px;
    padding: 14px;



}
.footer{
    margin-top: 1rem;
    padding: 1rem;
    background-color: rgb(235, 195, 64);
    position: fixed;
    bottom: 0;
```

```
    left: 0;
    width: 100%;
    font-weight: 1000;
      font-family: Georgia, 'Times New Roman', Times, serif;
  }
  .footer:hover{
      background-color: rgba(240, 54, 224, 0.747);
      cursor: pointer;
      font-weight: 600;
      font-
fami-
ly: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Genev
a, Verdana, sans-serif;
  }
  .footer::before{
      background-color: darkseagreen;
  }
```

4. Peek

```
import React from "react" ;
import NavBar from "./StackNavBar" ;
const Peek = ()=>{
    const url = "https://miro.medium.com/max/790/1*1okwhewf5KCtIPaFib4XaA.gif";

    const elem = <>
    <NavBar/>
    <div className="StackContainer">
        <h1><center><i>Peek Operation</i> </center> </h1>
        </div>
```

```
      <div className="StackContainer">
      <h2><b><center>STACK OPERATION (POP) ARE PERFORMED</center></b
></h2>

      </div>
      <div className="imgclass">
      <img className="arimg" src= {url}alt="al"></img>
      </div>
   </>
   return elem ;
}
export default Peek ;
```

5. Pop

```
import React from "react" ;
import NavBar from "./StackNavBar" ;
import Data from "./StackData/Data.json";
import "./Stack.css";

function NewlineText(props) {
   const text = props.text;
   return text.split('\n').map(str => <p>{str}</p>);
  }


const Pop = ()=>{
   const url = "https://gochronicles.com/content/images/2021/05/Stack2.gif";

   const elem = <>
   <NavBar/>
   <div className="StackContainer">
      <h2> <i> {Data[6].content}</i> </h2>
```

```
      </div>

    <div className="About">
    <br/>
    <NewlineText text = {Data[6].data} />
    <br/>
    <img className="Img" src= {url}alt="al"></img>
    </div>
    <br/>
    <br/>

    <div className ="About">
    <br/>

    <br/>
    </div>
    <br/>
    <div className ="About">
    <br/>


    <br/>
    </div>


    </>
    return elem ;
}
export default Pop ;
```

5.Push

```
import React from "react" ;
```

```
import SNavBar from "./StackNavBar" ;
import Data from "./StackData/Data.json";
import "./Stack.css";
import Img from "./Stack_Images/Stack_Array_Push.png" ;
import Img2 from "./Stack_Images/Stack_Linked_Push.png" ;



function NewlineText(props) {
   const text = props.text;
   return text.split('\n').map(str => <p>{str}</p>);
 }


const Push = ()=>{
   const url = "https://gochronicles.com/content/images/2021/05/Stack1.gif";

   const elem = <>
   <SNavBar/>
   <h2 className="StackContainer" >
       {Data[5].content}
   </h2>
   <div className="About">
   <NewlineText text = {Data[5].data} />
   <br/>
   <img className="Img" src= {url}alt="al"></img>
   </div><br/>
   <h2 className="StackContainer">
      <NewlineText text= {Data[5].content2} />
   </h2>
   <br/>
   <div className="About" >
   <NewlineText text = {Data[5].content5}/>
   <br/>
```

```
   <img src= {Img} className="Cont_Img" alt="array_pussh"/>

  </div>
  <br/>
  <h2 className="StackContainer">
  <NewlineText text=  {Data[5].content4} />
  </h2>
  <br/>
  <div className="About">
  <NewlineText text = {Data[5].content3} />
  <br/>
  <img src={Img2} className="Cont_Img" alt="push" />
  </div>


   </>
   return elem ;
}
export default Push ;
```

6. CSS

```
.StackContainer {
  margin-top: 30px;
  margin-left: 270px;
}

h1 {
  color: black;
  font-weight: 500;
  text-align: center;
  text-justify: auto;
  /*font-style: italic;*/
```

```
}

.Img {
  width: 100%;
  height: 235px;
  border-top-right-radius: 12px;
  border-top-left-radius: 12px;
  margin-top: 5px;
  margin-bottom: 5px;
}

.About {
  margin-left: 270px;
  background-color:  #50C878 ;
  padding: 25px 35px 35px 35px;
  border-style: groove;
  border: 2px solid grey;

  border-radius: 21px;
  width: 70%;
  font-
fami-
ly: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-
serif;
  font-size: 22px;
  box-sizing: border-box;
}
.Cont_Img{
  width: 100%;
  height: 400px;
  border-top-right-radius: 12px;
  border-top-left-radius: 12px;
  border-bottom-right-radius: 12px;
```

```
  border-bottom-left-radius: 12px;
  margin-top: 5px;
  margin-bottom: 5px;

}
```

7. JSX

```
import React from "react";
import "./Stack.css";
import StackNavBar from "./StackNavBar";
import Data from "./StackData/Data.json";

const url = "https://prmoreira23.github.io/assets/stack-data-structure.gif";

function NewlineText(props) {
    const text = props.text;
    return text.split('\n').map(str => <p>{str}</p>);
  }
const Stack = () => {
    const elem = <>
        <StackNavBar/>
        <h2 className="StackContainer" >
         <NewlineText text = {Data[0].content} />

        </h2>
        <p className = "About">

        {Data[0].about}
        <br/>
        <br/>
        <NewlineText text = {Data[0].data} />
```

```
      <br/>
      <img src ={url} className= "Img" alt="getImage" >


      </img>
      <br/>
      <br/>
      {Data[1].content}
      <br/>
      <br/>
      <NewlineText text = {Data[1].data}/>
      <br/>
      <br/>
      {Data[2].content}
      <br/>
      <br/>
      <NewlineText text = {Data[2].data}/>


      </p>
      </>


      return elem;
}
      export default Stack;
```

8. StackNavBar (CSS)

```
.StUl {
  list-style-type: none;
  margin-top: 2px;
```

```
  margin-right: 4px;
  padding: 0;
  width: 250px;
  background-color: white;
  position: absolute;
  height: 300%;
  overflow: auto;
}

.StLink {
  display: block;
  color: #000;
  padding: 8px 16px;
  text-decoration: none;
  font-size: 120%;
  font-weight: 500;
}


/* Change the link color on hover */



.StLink:hover {
  background-color: #555;
  color: white;
  border-radius: 30px;
  background-position: 0% 30%;
}

.stLink.active {
  background-color: #04AA6D;
  color: white;
}
```

```
h2 {
  padding: 8px 16px;
  color: rgb(233, 23, 65);
  font-size: 45px;
}

.Selected {
  color: /* #DC143C;*/
  white;
  background: grey
  /*linear-
gradient(to right, green 50%, green 50%, #AAAAAA 82%, #AAAAAA 80%)*/
  ;
  font-family: Verdana, Geneva, Tahoma, sans-serif;
}
```

9. Stack NavBar (JSX)

```
import React from "react";
import "./StackNavBar.css";
import { NavLink } from "react-router-dom";
const StackNavBar = () => {
   const elem = <>
      <ul className="StUl">
         <h2><center>Stack</center></h2>
         <li><NavLink className="StLink" to="/stack">About</NavLink></li>
         <li><NavLink className="StLink" activeClassName="Selected" to="/stack/arr
ay">Array Representation</NavLink></li>
         <li><NavLink className="StLink" activeClassName="Selected" to="/stack/lin
ked">Linked Representation</NavLink></li>
```

```
        <li><NavLink className="StLink" activeClassName="Selected" to="/stack/push">Push</NavLink></li>
        <li><NavLink className="StLink" activeClassName="Selected" to="/stack/pop">Pop</NavLink></li>


        <li><NavLink className="StLink" activeClassName="Selected" to="/stack/peek">peek</NavLink></li>
        <li><NavLink className="StLink" activeClassName="Selected" to="/stack/code">Complete Program</NavLink></li>
      </ul>

   </>
   return elem;
}
export default StackNavBar;
```

10. Data (JSON)

```
[
 {
  "id" :"1" ,
  "content": "Stack Introduction :",
  "about" : "Stack is a Linear Data Structure which works on : "  ,
  "da-
ta":"LIFO ( Last in First out ) Principal \n LIFO principal says that the element which is
 inserted at last will be deleted first from the stack."
 },
 {
  "id" :"2" ,
  "content":" Applications :  " ,
```

```
   "da-
ta":"1.  Used in backtracking problems \n 2. Used to check parenthesis matching in an e
xpres-
si-
on \n 3. Execution of SubPrograms \n 4. Supports Recursion \n 5. JVM( Java Virtual M
achine ) uses stack "

  },
  {
   "id" :"3" ,
   "content":"Practical Applications : " ,
   "da-
ta" : "1. Undo operations are possible by the help of Stack \n 2. To keep track of most re
cently used Tabs/file"
  },
  {
   "id" :"4" ,
   "content" : "Array Representation of Stack : "  ,
   "data" : "Stack can be Implemented by using an Array : " ,
   "da-
ta2" : "Suppose we have to create a stack which store integer type elements \n then an a
rray of integer type is taken \n and and another variable is taken which points to top of t
he stack ",
   "content2" :"Drawbacks of Array Representation :",
   "data3" : "Self refrential structure of Stack Node which stores integer type data :-  ",
   "data4" : "Suppose we taken the array arr \n and top variable as top " ,



   "da-
ta5" : "The Size of an array is fixed \n While, in a stack, there is no fixed size since the
size of stack changed with the number of elements pushed or popped from it. \n This dr
awback is rectified in linked representation  "
```

```
 },
 {
  "id" :"5" ,
  "content" : "Linked Representation of Stack : ",
  "data" : ""
 },
 {
  "id" :"6" ,
  "content" : "Push Operation : ",
  "da-
ta" : "Push operation refers to inserting an element in the stack. \n Since there's only on
e position at which the new element can be inserted i.e. Top of the stack, \n  the new ele
ment is inserted at the top of the stack." ,
   "content4":"Push operation \n( In Array Representation)  :" ,
  "content2":"Push operation \n( In Linked Representation)  :",
  "con-
tent3":"Push operation when a Stack is created by using an Array \n Here arr represents
the array\n top represents the variable which points to top \n the new element is inserted
 at the top \n and the value of top is incremented by one  " ,
   "con-
tent5":"Push operation when a Stack is created by using list \n here memory is allocated
 to newly created pointer of stack type \n then, The top started pointing the newly create
d object "
 },
 {
  "id":"7" ,
  "content" :"Pop Operation",
  "da-
ta" :"Pop operation refers to deleting an element from the top of the stack . \n After dele
tion if stack is not empty \n the value of Top is decreased by 1",
   "content2" :"Pop operation \n ( IN Linked Representation) : " ,
   "content3" :"Pop operation \n ( In Array Representation ) "
```

```
}
```

1. Contents (JSX)

```
import React from "react" ;
const Contents = (props) => {
   const elem = <div className="Compp">

      <div className="cards">
         <div className="card">
            <img src={props.Link} alt="MYIMG"
             className="cardImg"/>
            <div className="cardInfo">
            <h3 className="cardTitle">{props.Title}</h3>
               <span className="cardCategory"> {props.About}  </span>

            </div>
         </div>
      </div>

   </div>
   return elem ;
}
export default Contents ;
```

2. content (css)

```
* {
  box-sizing: border-box;
```

```
  padding: 0;
  margin: 0;
}

body {
  background-color: whitesmoke;
}

.cards {
  width: 100%;
  height: auto;
}

.card {
  margin-left: 6%;
  margin-right: 6%;
  margin-top: 3%;
  transition: all 0.4s cubic-beizer(0.175, 0.885, 0, 1);
  background-color: orange;
  width: 21.25%;
  border-radius: 12px;
  box-shadow: 0px 13px 10px -7px rgba(0, 0, 0, 0.1);
  float: left;
}

.card:hover {
  box-shadow: 0px 30px 18px -8px rgba(0, 0, 0, 0.3);
  transform: scale(1.06, 1.08);
}

.cardImg {
  width: 100%;
  height: 235px;
```

```
  border-top-right-radius: 12px;
  border-top-left-radius: 12px;
}

.cardInfo {
  border-bottom-left-radius: 12px;
  border-bottom-right-radius: 12px;
  padding: 16px 24px 24px 24px;
  background-color: #fff;
}

.cardCategory {
  font-family: "Raleway", sans-serif;
  text-transform: uppercase;
  font-size: 13px;
  letter-spacing: 2px;
  font-weight: 500;
  color: #868686;
  background-color: #fff;
}

.cardTitle {
  margin-top: 5px;
  margin-bottom: 10px;
  font-family: "Raleway", sans-serif;
  color: black;
  background-color: #fff;
  text-transform: capitalize;
}
```

```
import * as React from "react";
```

```
import { useForm } from "react-hook-form";

export default function App() {
  const { register, handleSubmit } = useForm();
  const onSubmit = (data) => alert(JSON.stringify(data));

  return (
    <form onSubmit={handleSubmit(onSubmit)}>
      <input {...register("firstName", { required: true })} placeholder="First name" />

      <input {...register("lastName", { minLength: 2 })} placeholder="Last name" />

      <select {...register("category")}>
        <option value="">Select...</option>
        <option value="A">Category A</option>
        <option value="B">Category B</option>
      </select>

      <input type="submit" />
    </form>
  );
}
```

4.3 Src/Data

1. Contents Data

```
const ContentsData = [
    {
        Link:"https://i2.wp.com/blog.contactsunny.com/wp-
content/uploads/2019/12/photo-1559944554-
fb295d391db7.jpeg?resize=1400%2C850&ssl=1",
        Title:"Linked List" ,
        Description:"Explore Linked List"
    }
    ,{
        Link : "https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcRGs7WcpE7uU3uGBeA_Fb1_zrNL65lda_-H-
Q&usqp=CAU" ,
       Title: "Stack" ,
```

```
    Description :"Explore Stack"

  } ,
  {
    Link: "https://koenig-
media.raywenderlich.com/uploads/2017/06/HeapPriorityQueue-feature.png" ,
    Title:"Queue",
    Description :"Explore Queue"

  },
  {
    Link:"https://neverstopcodingblog.files.wordpress.com/2017/07/tree.jpg?w=450&
h=300&crop=1",
    Title:"Binary Search Tree" ,
    Description:"Explore Binary Search Tree"
  },
]
export default ContentsData ;
```

4.4 Src/NavBar

1. NavBar (jsx)

```
import "./NavBar.css" ;
import {NavLink} from "react-router-dom";
const NavBar = ()=>{
  const elem=<div className="NavContainer">

<nav className="Lis">
 <li><p className="Home">DS_Arena</p></li>
 <li><a className="Cont" href="#Login">LOGIN</a></li>
 <li><NavLink className="Cont"  to="/">Home</NavLink></li>
 <li><a className="Cont" href="#about">About</a></li>
```

```
 <li><NavLink className="Cont"  to="/candidate">Candidates</NavLink></li>
 <li><a className="Cont" href="#blog">Blog</a></li>
<li><a className="Cont" href="#contact">Contact</a></li>
</nav>


  </div>
  return elem;
}
export default NavBar ;
```

2. NavBar(CSS)

```
.NavContainer {
 width: 100%;
 font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
 font-style: italic;
 visibility: unset;
 display: block;
}

.Lis {
 list-style-type: none;
 margin: 0;
 padding: 0;
 overflow: hidden;
 background-color: black;
 opacity: 0.9;
}

.Cont {
 display: block;
 color: white;
 font-weight: 600;
 text-align: center;
 padding: 15px 18px;
 text-decoration: none;
 float: right;
 padding-right: 30px;
}

.Home {
 display: block;
 font-weight: 1000;
 color: white;
```

```
  font-size: 130%;
  padding-top: 10px;
  text-align: center;
  padding-left: 100px;
  font-family: cursive;
  letter-spacing: 4px;
  float: left;
}


/* Change the link color to #111 (black) on hover */

.Cont:hover {
 /* background-color: rgb(172, 150, 150);*/
  display: inline-block;
  position: relative;
  -webkit-transition: all 200ms ease-in;
  -webkit-transform: scale(1);
  -ms-transition: all 200ms ease-in;
  -ms-transform: scale(1);
  -moz-transition: all 200ms ease-in;
  -moz-transform: scale(1);
  transition: all 200ms ease-in;
  transform: scale(1);
  background-color: grey;
  box-shadow: 0px 0px 200px black;
  z-index: -1;
  -webkit-transition: all 200ms ease-in;
  -webkit-transform: scale(1);
  -ms-transition: all 200ms ease-in;
  -ms-transform: scale(1);
  -moz-transition: all 200ms ease-in;
  -moz-transform: scale(1);
  transition: all 200ms ease-in;
  transform: scale(1);
}
```

4.5 AllCont (jsx)

```
import React from "react";
import Content from "./Components/Contents";
import Data from "./Data/ContentsData";
import {NavLink} from "react-router-dom"  ;
import "./Components/Contents.css" ;

const AllCont = () => {
   const elem = <>

       <NavLink to="/list">
       <Con-
tent Link={Data[0].Link} Title={Data[0].Title} About={Data[0].Description} />
       </NavLink>
       <NavLink to="/stack">
       <Con-
tent Link={Data[1].Link} Title={Data[1].Title} About={Data[1].Description} />
       </NavLink>
       <NavLink to="/queue">
       <Con-
tent Link={Data[2].Link} Title={Data[2].Title} About={Data[2].Description} />
       </NavLink>
       <NavLink to="/tree">
       <Con-
tent Link={Data[3].Link} Title={Data[3].Title} About={Data[3].Description} />
       </NavLink>

   </>
   return elem;
}
```

```
export default AllCont;
```

4.6 App(jsx)

```
/*
   All the Components which are rendered by clicking from
   AllCont document are rendered from this page
*/
import React from "react";
import Stack from "./Components/Stack/Stack";
import AllCont from "./AllCont";
import NavBar from "./NavBar/NavBar";
import Push from "./Components/Stack/Push";
import Pop from "./Components/Stack/Pop";
import Peek from "./Components/Stack/Peek";
import { Switch, Route } from "react-router-dom";
import ArrayR from "./Components/Stack/ArrayRepresentation";
import LinkedR from "./Components/Stack/LinkedRepresentation";
import Code from "./Components/Stack/CompleteProgram";


const App = () => {
   const elem = <div className="Divv">
      <NavBar />
      <Switch>
         <Route exact path="/" component={AllCont} />
         <Route exact path="/Stack" component={Stack} />
         <Route exact path="/stack/push" component={Push} />
         <Route exact path="/stack/pop" component={Pop} />
         <Route exact path="/stack/peek" component={Peek} />
         <Route exact path="/stack/array" component={ArrayR} />
         <Route exact path="/stack/linked" component={LinkedR} />
         <Route exact path="/stack/code" component={Code} />
```

```
      </Switch>
    </div>
    return elem;
}
export default App;
```

4.7. Index (js)

```
import React from "react";
import ReactDOM from "react-dom";
import App from "./App";
import {BrowserRouter} from"react-router-dom" ;
ReactDOM.render(
<BrowserRouter>
<App/>
</BrowserRouter>,

  document.getElementById("root")
)
```

CHAPTER 5 TESTING

5.1 INTRODUCTION

Objectives

Thefollowing arethetesting objectives:

- Testingisaprocess ofexecutingaprogram withtheintentoffind-inganerror.
- Agoodtestcaseisonethathasahighprobability offindinganas-yet-undiscoverederror
- successfultestisonethatuncoversanasyetundiscoverederror.

Principles

Thebasicprinciplesthatguidesoftwaretesting areas follows:

- Alltests shouldbetraceabletocustomerrequirements.
- Testsshouldbeplannedlongbeforetestingbegins.
- Theparateprincipleappliesto softwaretesting.

Paretoprinciplestatest-hat80percentofallerrorsuncoveredduringtestingwilllikelybetraceableto 20 percentof all programcomponents.

Testingshouldbegin“inthesmall “andprogresstowardtest-ing“inthelarge.”Exhaustivetesting is notpossible.

5.2 LEVELOF TESTING

Therearedifferentlevelsoftesting

- UnitTesting
- IntegrationTesting
- SystemTesting

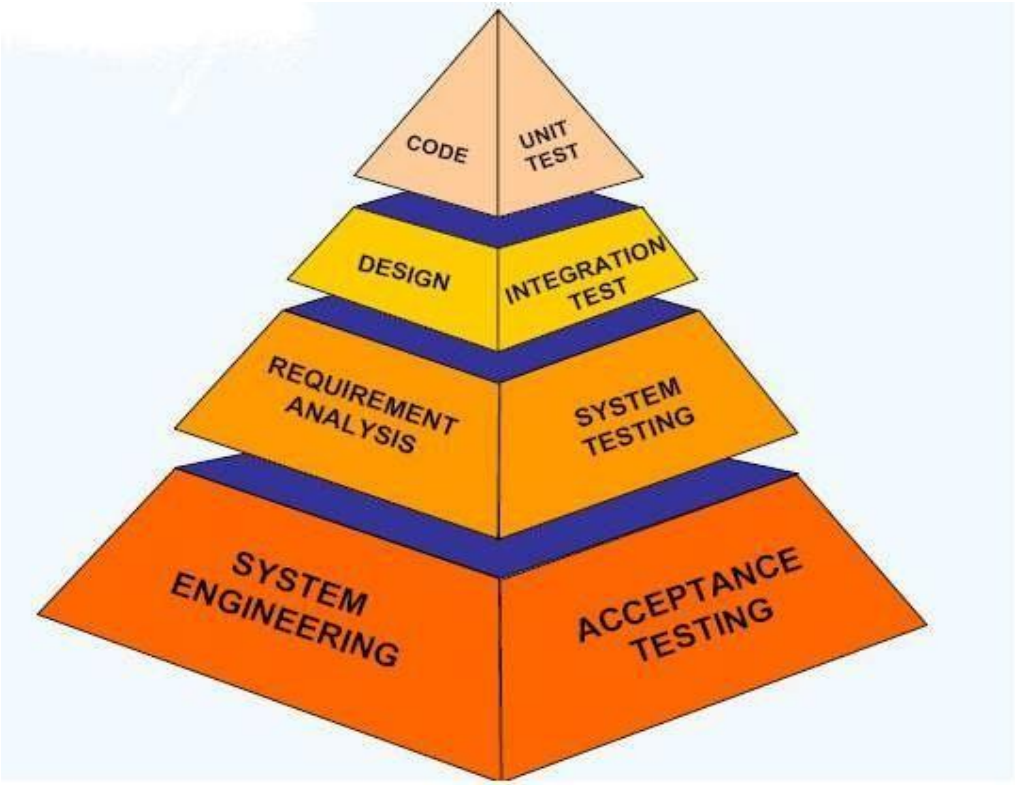

Figure 9 Testing Levels

Unittesting

Unit testing focuses verification effort on the smallest unit ofsoftware design, the module. The important control parts are tested to uncover with in theboundary of the module. The module interface is tested to ensure that the informationproperlyflows intoand out ofthe programunit and boundaryconditions aretestedtoensurethatthemodulesoperateproperlyatboundariesestablishedtolimit orrestrict

processing.Testdateisprovidedthroughtestingscreens.

Integrationtesting

Integrating testing is a systematic technique for constructing Program structure whileconductingteststouncovererrorassociateswithinterfacing.Theobjectiveistotakeunitmodulesand built aprogram structurethat has been directedby design.

- IntegrationTestingwilltestwhetherthemodulesworkwelltogether.
- Thiswillcheckwhetherthedesigniscorrect.
- Integrationcanbedonein4differentways:

Systemtesting

System testing is the process of testing the completed software as a part of theenvironment itwas created for. It is done to ensure that all the requirements specified by thecustomeraremet.Systemtesting involvesfunctionaltestingand performancetesting.

- SystemTestingwillcontain thefollowingtesting:

  FunctionalTesting.

  PerformanceTesting.

- FunctionTestingwilltesttheimplementation ofthebusinessneeds.
- Performance Testing will test the non-functional requirements of the system like the speed,loadetc.

Chapter6. Future scope

We are focused that in future we will consistantly work on this project and try to add the functionality described below:

- We specified Node Js in our technical stack we will add backend by the help of Node js.
- We try to make user interface more interactive and look and feel more better.
- A discussion section will be added so that users may interact there and discuss their issues.
- A login and SigUp functionality will be introduced
- A feedback section will be introduced
- If user wants improvement in any content then he can submit the imporoved version and after the approval of admin the changes will be shown in the orignal content.
- A quiz section will be introduces so that if any user want to clear the concepts of any topic then he would attemp the quiz to track his progress.

Chapter 7. Conclusion and Bibliography

7.1 Conclusion

The main objective of this project is to develop a single page web application which can help many of the willing students who wants to learn data structures but can't able to decide that from where they can start.

For this project we choose React and NodeJs(Future scope) in React we learn lots of new things and along with learning we also implement them in our project.

There is a saying that

Journey of thousand miles starts with a single step

And this project helped us to clear the fundamental concepots of React and in future we are focused to work consistantly on this project and improve it in every aspect and we will also add the remaining backend functionalities.

We are very thankful to our Mentor he always motivated us and Inspire us nto learn new things along with their practical implementation.

7.2 Bibliography

These website helped me alot in the development of this application :

1. https://reactjs.org/docs/getting-started.html
2. https://www.w3schools.com
3. https://www.thapatechnical.com
4. https://css-tricks.com
5. https://developer.mozilla.org/en-US/docs/Web/JavaScript

These links helped me to collect propers photos for my application :

1. https://i2.wp.com/blog.contactsunny.com/wp-content/uploads/2019/12/photo-1559944554-fb295d391db7.jpeg?resize=1400%2C850&ssl=1

2. https://encryptedtbn0.gstatic.com/images?q=tbn:ANd9GcRGs7WcpE7uU3uGBeA_Fb1_zrNL65lda_-H-Q&usqp=CAU

3. https://koenig-media.raywenderlich.com/uploads/2017/06/HeapPriorityQueue-feature.png

4. https://neverstopcodingblog.files.wordpress.com/2017/07/tree.jpg?w=450&h=300&crop=1

7.3 GITHUB LINK OF PROJECT REPOSITORY:

https://github.com/shivamnerwal1998/DS_Arena.git