

TASK 7.2D(SIT313)

Submitted by:- Jeetasha(2210994796)

- Github Repository Link :-

<https://github.com/JeetashaArora/SIT313-/tree/main/Task%207.2D>

- YouTube Video Link :-

<https://youtu.be/zNj4xY8n2uA>

- Code Screenshots:-
- Article Component

```
1 |
2 import React, { useEffect } from 'react'
3 import Title from "./Title"
4 import Textbox from "./Textbox"
5 import Header from "./Header"
6 import PostButton from "./Button"
7 import "./Article.css"
8 import FontAwesomeIcon from 'react-fontawesome'
9 import { useState } from 'react'
10 import ReactTagInput from '@pathofdev/react-tag-input'
11 import "@pathofdev/react-tag-input/build/index.css"
12 import { getDownloadURL, ref, uploadBytes } from 'firebase/storage'
13 import { db, storage } from '../utility/firebase'
14 import { doc, addDoc, collection, updateDoc, deleteDoc, getDoc, getDocs } from 'firebase/firestore'
15
16
17 export default function Article() {
18   const [title, settitle] = useState(" ")
19   const [abstract, setabstract] = useState(" ")
20   const [article, setarticle] = useState(" ")
21   const [tags, settags] = useState([])
22   const [file, setfile] = useState(" ")
23   //referencing the database
24   const dbref = collection(db, "Articles")
25   //Storing images in firebase storage
26   const uploadImageInStorage = async (imageFile) => {
27     if (!imageFile) {
28       // Handle the case where no file is selected
29       console.error("No file selected");
30       return null;
31     }
32
33     try {
34       const storageRef = ref(storage, 'images/' + imageFile.name);
35       const snapshot = await uploadBytes(storageRef, imageFile);
36       const downloadUrl = await getDownloadURL(snapshot.ref);
37       return downloadUrl;
38     } catch (error) {
39       console.error("Error uploading image:", error);
40       return null;
41     }
42   };
43 }
```

- Button Component

```
1 import React from 'react'
2 import { Button } from 'semantic-ui-react'
3
4 const ButtonExampleFloated = () => (
5   <div className='container btn right 10'>
6     <Button color="teal" floated='right' >Post</Button>
7   </div>
8 )
9
10 export default ButtonExampleFloated
```

- Navbar Component

```
1 import Container from 'react-bootstrap/Container';
2 import Nav from 'react-bootstrap/Nav';
3 import Navbar from 'react-bootstrap/Navbar';
4 import NavDropdown from 'react-bootstrap/NavDropdown';
5 import { Link } from 'react-router-dom';
6
7 function BasicExample() {
8   return (
9     <Navbar expand="lg" className="bg-body-tertiary">
10       <Container>
11         <Navbar.Brand href="#home">Discussion Forum</Navbar.Brand>
12         <Navbar.Toggle aria-controls="basic-navbar-nav" />
13         <Navbar.Collapse id="basic-navbar-nav">
14           <Nav className="me-auto">
15             <Nav.Link><Link to="/">Home</Link></Nav.Link>
16             <Nav.Link><Link to="/find">FindQuestion</Link></Nav.Link>
17           </Nav>
18         </Navbar.Collapse>
19       </Container>
20     </Navbar>
21   );
22 }
23
24 export default BasicExample;
```

- PostType Component

```
1
2 import React, { Component, useState } from 'react'
3 import { Form, Radio } from 'semantic-ui-react'
4 import Question from './Question';
5 import Article from './Article';
6 const NewPost={()=>{
7   const [type,setType]=useState("question");
8   const handlePostChange=(e,{value})=>setType(value)
9   return (
10     <Form className='form container'>
11       {/* <Header className=" upline " text="New Post"></Header> */}
12       <h5 className='bg_header'>New Post</h5>
13       <div className='inlineTextbox'>
14         <Form.Field>
15           Selected one option
16         </Form.Field>
17         <br></br>
18         <Form.Field>
19           <Radio
20             label='Question'
21             name='post'
22             value='question'
23             checked={type === 'question'}
24             onChange={handlePostChange}
25           />
26         </Form.Field>
27         <Form.Field>
28           <Radio
29             label='Article'
30             name='post'
31             value='article'
32             checked={type === 'article'}
33             onChange={handlePostChange}
34           />
35         </Form.Field>
36       </div>
37       {type==="question"? <Question/>:<Article/>}
38     </Form>
39   )
40 };
41 export default NewPost;
42
```

- QuestionCard Component

```
1
2 import { doc,deleteDoc } from 'firebase/firestore';
3 import React, { useState } from 'react'
4 import FontAwesome from 'react-fontawesome';
5 import { Link } from 'react-router-dom';
6 import { Card, Icon } from 'semantic-ui-react'
7 import { db } from '../utility/firebase';
8
9 const QuestionCard = ({ title, problem, tags ,date,id}) => {
10   console.log(problem,"problemCheck");
11   const [isExpanded, setIsExpanded] = useState(false);
12
13   const handleReadMoreClick = () => {
14     setIsExpanded(!isExpanded);
15   };
16   const handleDelete = async (questionId) => {
17     try {
18       // Reference the document to be deleted
19       const questionDocRef = doc(db, 'Questions', questionId);
20
21       // Delete the document from Firestore
22       await deleteDoc(questionDocRef);
23
24       // Update the state to remove the deleted question
25       const updatedQuestions = questions.filter((question) => question.id !== questionId);
26       setQuestions(updatedQuestions);
27     } catch (error) {
28       console.error('Error deleting question: ', error);
29     }
30   }
31
32   return (
33     <Card fluid>
34       <Card.Content >
35         <Card fluid color='red' header={title} />
36         <Card.Meta>Tags: {tags}</Card.Meta>
37         <Card.Meta>Date: {date}</Card.Meta>
38         {isExpanded ? (
39           <Card.Description>{problem}</Card.Description>
40         ) : (
41           <Card.Description>{problem?problem.substring(0, 50):""}</Card.Description>
42
43         ) : (
44           <Card.Description>{problem?problem.substring(0, 50):""}</Card.Description>
45         )}
46       </Card.Content>
47       <div>
48         <button className='btn btn-read' onClick={handleReadMoreClick}>
49           {isExpanded ? 'Read Less' : 'Read More'}
50         </button>
51       </div>
52       <div style={{ float: 'right', cursor: 'pointer' }}>
53         <Icon name='trash' style={{ margin: '15px' }} size='big' onClick={()=>handleDelete(id)} />
54       </div>
55     </Card.Content>
56   </Card>
57 );
58 };
59
60 export default QuestionCard;
```

- **Question Component**

```
1 import React, { useState } from 'react'
2 import Title from './Title'
3 import Textbox from './Textbox'
4 import PostButton from './Button'
5 import Header from './Header'
6 import { doc, addDoc, collection, updateDoc, deleteDoc, getDoc, getDocs } from 'firebase/firestore'
7 import { db, storage } from '../utility/firebase'
8 import "react-bootstrap"
9 import {v4 as uuidv4} from "uuid"
10
11 export default function Question() {
12   const [title, settitle]=useState(" ")
13   const [problem, setproblem]=useState(" ")
14   const [tags, settags]=useState([])
15   const [currentDate, setCurrentDate] = useState(new Date().toISOString().split('T')[0]);
16
17   const dbref = collection(db, "Questions")
18   const savedata = async (title, problem, tags) => {
19     const id=uuidv4();
20     try{
21       const addData = await addDoc(dbref, {
22         Title: title,
23         Problem:problem,
24         Tags: tags,
25         Id: id,
26         Date: currentDate,
27       });
28     }
29     if (addData) {
30       alert("Posted successfully");
31       console.log(currentDate)
32     }
33   }
34   catch (error) {
35     console.error(error);
36   }
37 }
38
39 const handleBrowseClick = () => {
40   document.getElementById("fileinput").click();
41 }
```

```
41 return (
42   <div >
43     <div >
44       <Header text="What do you want to ask or share"></Header>
45       <div className='inlineTextbox'>
46         <span className=''><pre>Title </pre></span> <input placeholder="Start your question with how what why etc. " value={title} className="QuesTitle" cols="3902" r
47         <input type="text" value={title} placeholder="Title" />
48       </div>
49       <div></div>
50       <br></br>
51       Describe your problem:
52       <textarea className="container mb-3" value={problem} heading="Describe your problem" rows="18" cols="110" onChange={(e)=>{
53         console.log(e.target.value,"Problem");setproblem(e.target.value)}}></textarea>
54       <br></br>
55       <div className='inlineTextbox'>
56         <span><pre>Tags </pre></span><input value={tags} placeholderText="Please add upto 3 tags to describe what your question is about e.g. Java"
57         <input type="text" value={tags} placeholder="Tags" />
58       </div>
59       <br></br>
60       <button className='btn-add' onClick={()=>savedata(title,problem,tags)}>Post</button>
61     </div>
62   </div>
63 )
64 }
65 }
```

- QuestionList Component

```
1 import React, { useState } from 'react';
2 import QuestionCard from "../Question Card"
3
4
5
6
7
8 const QuestionList = ({ questions }) => {
9   const [filterCriteria, setFilterCriteria] = useState({
10     date: '',
11     tag: '',
12     title: '',
13   });
14
15   const [filteredQuestions, setFilteredQuestions] = useState(questions);
16
17   const filterQuestions = () => {
18     const { date, tag, title } = filterCriteria;
19
20     // Perform filtering based on date, tag, and title
21     const filtered = questions.filter((question) => {
22       return (
23         (date === '' || question.Date.includes(date)) &&
24         (tag === '' || question.Tags.includes(tag)) &&
25         (title === '' || question.Title.toLowerCase().includes(title.toLowerCase()))
26       );
27     });
28
29     // Update the filtered questions state
30     setFilteredQuestions(filtered);
31   };
32 }
```

```
return (
  <div>
    <h2>Question List</h2>
    /* Add filter inputs here */
    <div>
      <input
        type="text"
        placeholder="Date"
        value={filterCriteria.date}
        onChange={(e) => setFilterCriteria({ ...filterCriteria, date: e.target.value })}
      />
      <input
        type="text"
        placeholder="Tag"
        value={filterCriteria.tag}
        onChange={(e) => setFilterCriteria({ ...filterCriteria, tag: e.target.value })}
      />
      <input
        type="text"
        placeholder="Title"
        value={filterCriteria.title}
        onChange={(e) => setFilterCriteria({ ...filterCriteria, title: e.target.value })}
      />
      <button onClick={filterQuestions}>Filter</button>
    </div>
    {(filterCriteria.date === '' && filterCriteria.tag === '' && filterCriteria.title === '') ?
      questions.map((question) => (
        <QuestionCard
          key={question.id}
          title={question.Title}
          problem={question.Problem}
          tags={question.Tags}
          date={question.Date}
          id={question.id}
        />
      )) :
      <div>No questions found</div>
    }
  </div>
)
```

```

70
71     }):
72     filteredQuestions.map((question) => (
73         <QuestionCard
74             key={question.id}
75             title={question.Title}
76             problem={question.Problem}
77             tags={question.Tags}
78             date={question.Date}
79             id={question.id}
80         />
81     ))
82 }
83
84
85 </div>
86 );
87 };
88
89 export default QuestionList;
90

```

- FindQuesPage Component

```

1  import React, { useEffect, useState } from 'react'
2  import QuestionList from '../components/QuestionList'
3  import { collection, onSnapshot } from 'firebase/firestore';
4  import { db } from '../utility/firebase';
5
6  const FindQues = () => {
7      const [loading, setloading] = useState(true);
8      const [questions, setQuestions] = useState([]);
9
10     useEffect(() => {
11         const unsub = onSnapshot(
12             collection(db, "Questions"),
13             (snapshot) => {
14                 let list = []
15                 snapshot.docs.forEach((doc) => {
16                     list.push({id: doc.id, ...doc.data()});
17                 })
18                 setQuestions(list)
19             }, (error) => {
20                 console.log(error);
21             })
22         );
23         return () => {
24             unsub();
25         }
26     }, []);
27     console.log("questions", questions)
28     return (
29         <div>
30             <h2>Find the Questions!!</h2>
31             <QuestionList questions={questions}></QuestionList>
32         </div>
33     )
34 }
35 export default FindQues

```

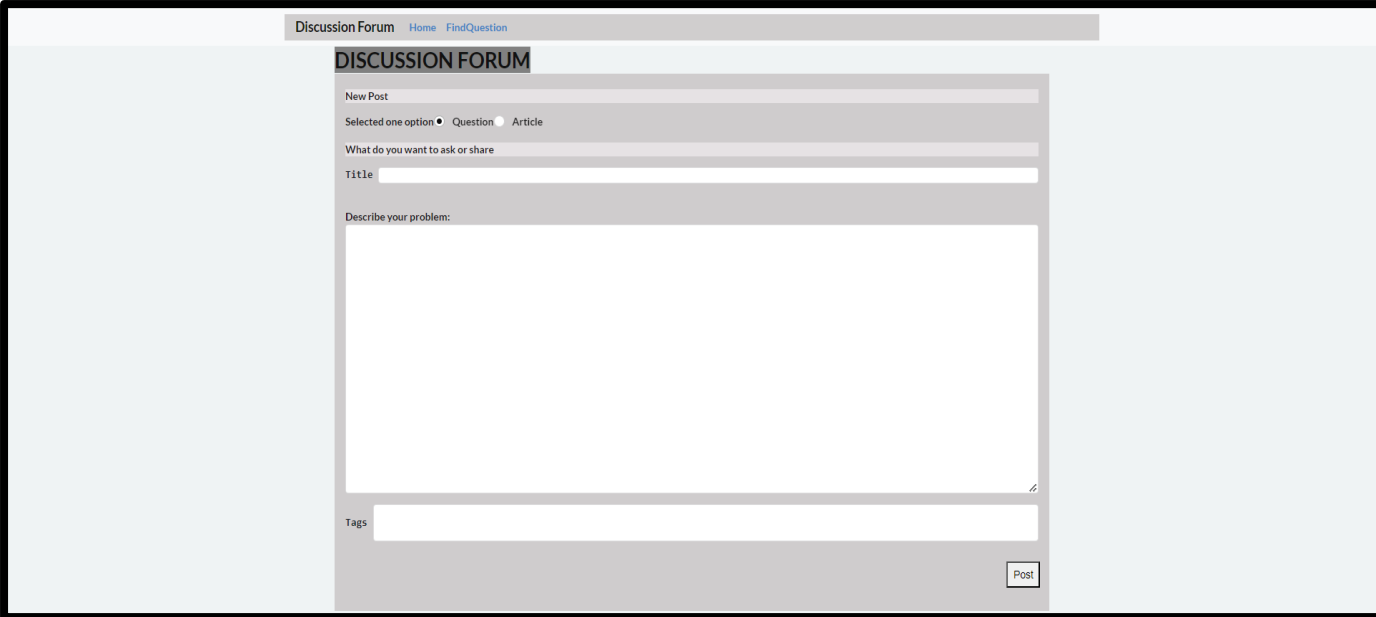
- **Firebase.js**

```
1 import {initializeApp} from 'firebase/app'
2 import {collection, getDocs, getFirestore, query} from 'firebase/firestore'
3 import {getStorage} from 'firebase/storage'
4 import { useState } from 'react';
5 const firebaseConfig = {
6   apiKey: "AIzaSyBBtdwSMfFaGA_KENeYQ74TPfhI2LR5yUo",
7   authDomain: "react-post-e7d02.firebaseio.com",
8   projectId: "react-post-e7d02",
9   storageBucket: "react-post-e7d02.appspot.com",
10   messagingSenderId: "298731947835",
11   appId: "1:298731947835:web:f638c4d5550d4f4838d91e"
12 };
13 const app=initializeApp(firebaseConfig);
14 const db=getFirestore(app);
15 const storage=getStorage(app);
16 export{db,storage};
17 export const excerpt=(str,count)=>{
18   if(structuredClone.length>count)
19   {
20     str=str.substring(0,count)+"...";
21   }
22   return str;
23 }
24 export async function fetchQuestions(e)
25 {
26   const [fetchedQuestions,setAllQuestions]=useState([])
27   e.preventDefault();
28   db.collection("Questions")
29   .get()
30   .then((snapshot)=>{
31     if(snapshot.docs.length>0)
32     {
33       snapshot.docs.forEach((doc)=>{
34         setAllQuestions((prev)=>{
35           return[...prev,doc.data()];
36         })
37       })
38     }
39   })
40   console.log(fetchedQuestions)
41 }
42
43
```


- App.js

```
1 import logo from './logo.svg';
2 import './App.css';
3 import Header from './components/Header.jsx'
4 import PostType from './components/PostType';
5 import 'bootstrap/dist/css/bootstrap.min.css';
6 import './components/Article.css'
7 import "react-toastify/dist/ReactToastify.css"
8 import HeaderImage from './components/HeaderImage';
9 import { Route ,BrowserRouter,Routes} from 'react-router-dom';
10 import FindQues from './pages/FindQues';
11 import NotFound from './pages/NotFound';
12 import { ToastContainer } from 'react-bootstrap';
13 import Navbar from './components/Navbar';
14 function App() {
15   return (
16     <div>
17       <Navbar></Navbar>
18       <div className='alignment'>
19         <h1 className='first text-center'>DISCUSSION FORUM</h1>
20         <ToastContainer/>
21         <Routes> (property) element: React.JSX.Element
22           <Route path="/" element={<PostType ques></PostType>}></Route>
23           <Route path="/find" element={<FindQues></FindQues>}></Route>
24           <Route path="*" element={<NotFound></NotFound>}></Route>
25         </Routes>
26       </div>
27     </div>
28   );
29 }
30 export default App;
31
```

- Screenshot of Questions Page



The screenshot shows a web application interface for a 'Discussion Forum'. At the top, there is a navigation bar with the text 'Discussion Forum' and two links: 'Home' and 'FindQuestion'. Below the navigation bar, the main heading 'DISCUSSION FORUM' is displayed. Underneath the heading, there is a section titled 'New Post'. This section includes a radio button selection for 'Question' (which is selected) and 'Article'. Below this, there is a text input field labeled 'What do you want to ask or share'. Further down, there is a label 'Title' followed by a text input field. Below the title field, there is a label 'Describe your problem:' followed by a large text area for input. At the bottom of this section, there is a label 'Tags' followed by a text input field. Finally, at the bottom right of the form, there is a 'Post' button.

- **Screenshot of Articles Page**

The screenshot shows the 'New Post' form on the 'Discussion Forum' page. The page has a header with 'Discussion Forum' and links to 'Home' and 'FindQuestion'. The form is titled 'DISCUSSION FORUM' and has a 'New Post' section. Below this, there are radio buttons for 'Question' (selected) and 'Article'. The form includes a 'What do you want to ask or share' section with a 'title' input field. Below the title is an 'Add an Image' section with a 'Choose file' button and a 'No file chosen' status. To the right of this is a 'Browse' button and an 'Upload' button. Below the image section is an 'Abstract' input field. Below the abstract is an 'Article Text' input field. At the bottom of the form is a 'Tags' section with a placeholder text 'Please add upto 3 tags to describe what your question is about e.g. Java'. A 'Post' button is located at the bottom right of the form.

- **Screenshot of Findquestions Page**

The screenshot shows the 'Find the Questions!!' page on the 'Discussion Forum'. The page has a header with 'Discussion Forum' and links to 'Home' and 'FindQuestion'. Below the header is a 'Question List' section. At the top of the list is a search bar with the text 'Find the Questions!!'. Below the search bar are four filter buttons: 'Date', 'Tag', 'Title', and 'Filter'. The list contains two questions. The first question is titled 'Fullstack' and has tags 'HTML' and 'Date: 2023-10-01'. The second question is titled 'react' and has tags 'React' and 'Date: 2023-10-01'. Both questions have a 'Read More' link and a trash icon. The page also has a 'New Post' button at the bottom left.