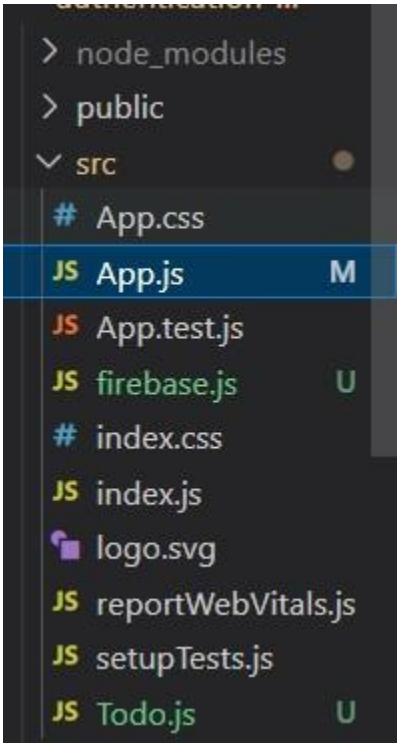
Week-11. Connecting our TODO React js Project with Firebase

We all can create applications but in realtime when we are building an application we have to store the user data some ware now a days best way to store is Firebase which can be integrated in react app

In this week we will learn how to connect our application to firebase

File Structure:



After creating the project make sure to install firebase dependencies:

Install it using npm install firebase

-Now we have mainly 3 pages
1.firebase.js
2.App.js
3.Todo.js

-.In firebase.js we will establish connection to our app and firebase
-In Todo,js we will write the code
And we will import it in to the App.js file

firebase.js

```
import firebase from 'firebase/compat/app'; import
'firebase/compat/auth';
import 'firebase/compat/firestore';

const firebaseApp =
firebase.initializeApp({     apiKey: "",
authDomain: "",    projectId: '',
storageBucket: '',    messagingSenderId: "",
appId: ':,    measurementId: ""
});

const db = firebaseApp.firestore();

export default db;
```

Note Replace the highlighted code with your firebase connection compoenents

You can get you own keys from firebase account for more details Take the
Reference of below video

https://www.youtube.com/watch?v=ad6IavyAHsQ

Todo.js

```
import { ListItem, List, ListItemAvatar, ListItemText, Button, Modal,
makeStyles } from '@material-ui/core' import
'./Todo.css';
import React, { useState } from 'react'; import
db from './firebase'



function Todo(props) {
  const [open, setOpen] = useState(false);
  const [input, setInput] = useState(props.todo.todo);

  const handleOpen = () => {
    setOpen(true)
  };

  const updateTodo = () => {
    // update to do with the new input text


db.collection('todos').doc(props.todo.id).set({
todo: input        }, { merge: true })

    setOpen(false);
  }

  return (
<>
        <div
open={open}
          onClose={e => setOpen(false)}
        >
          <div >
            <h1>I am a model</h1>
```

```
                    <input          placeholder={props.todo.todo}
            value={input} onChange={event => setInput(event.target.value)} />
                    <button onClick={updateTodo}>Update Todo</button>
                  </div>
                </div>
                <ul className='todo_list'>
                  <li>
                    <li      primary={props.todo.todo}
            secondary='Dummy deadline ⏰' />
                  </li>
                  <button onClick={e => setOpen(true)}>Edit</button>
                  <button                        onClick={event                        =>
        db.collection('todos').doc(props.todo.id).delete()}>❌DELETE
        ME</button>
                </ul>
              </>
            )
        }

        export default Todo
```

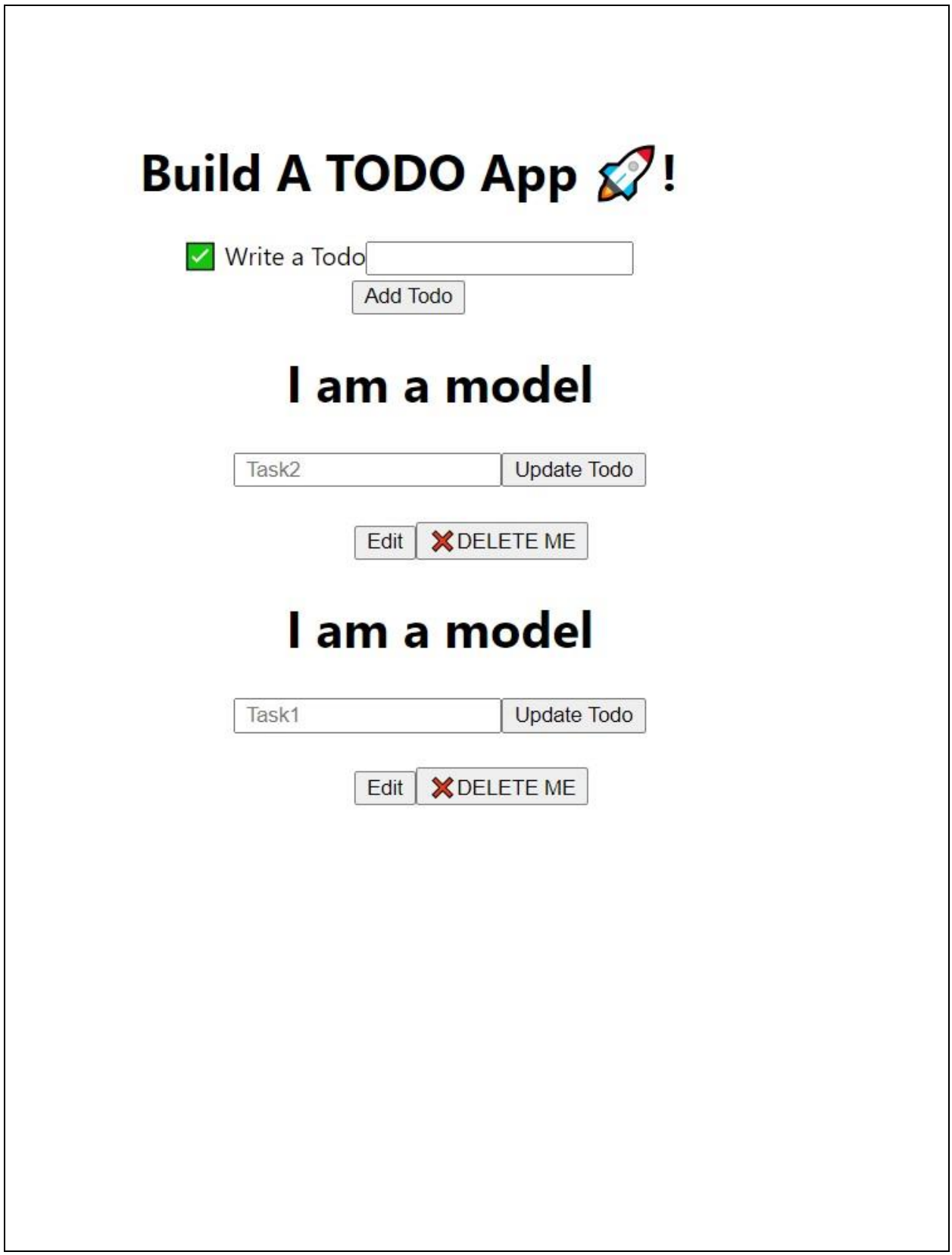
Now the last file App.js

```
import React, { useEffect, useState } from 'react';
import './App.css'; import Todo from './Todo';
import db from './firebase'
import firebase from 'firebase/compat/app'; import
'firebase/compat/auth';
import 'firebase/compat/firestore';

function App() {
  const [todos, setTodos] = useState([]);
```

```
  const [input, setInput] = useState('');

// when the upload, we need to listen to the database and fetch new todos as they get
added/remove useEffect(() => {
  // This code here... fires when the app.js lodes
  db.collection('todos').orderBy('timestamp', 'desc').onSnapshot(snapshot => {
    // console.log(snapshot.docs.map(doc => doc.data()));
    setTodos(snapshot.docs.map(doc => ({id: doc.id, todo: doc.data().todo})))
  })
}, []);

  const addTodo = (event) => {
    // this will fire off when we click the button
event.preventDefault(); //will stop the refresh

    db.collection('todos').add({
      todo: input,
      timestamp: firebase.firestore.FieldValue.serverTimestamp()
    })

    setTodos([...todos, input]);
    setInput(' ');  // clear up the input after clicking todo
console.log(todos)
  }

  return (
    <div className="App">
      <h1>Build A TODO App 🚀!</h1>

      <form>

        <form>
          <span>☑ Write a Todo</span>
          <input value={input} onChange={event => setInput(event.target.value)} />
</form>
```

```
                                                    onClick={addTodo}


      <button disabled={!input} type='submit'
variant="contained" color="primary">Add Todo</button>
</form>

     <ul>
       {todos.map(todo => (
         <Todo todo={todo}/>
         // <li>{todo}</li>
       ))}

       <li></li>
     </ul>

   </div>
  );
}

export default App;
```

OUTPUT


Build A TODO App 🚀!
Write a Todo
Add Todo
I am a model
Task2
Update Todo
Edit
DELETE ME
I am a model
Task1
Update Todo
Edit
DELETE ME