

SPRINT - 1

Date	29 October 2022
Team ID	PNT2022TMID01880
Project Name	Smart Waste Management System for Metropolitan Cities

OBJECTIVE:

Create a login portal to register the workers in system and to maintain the number of user. used **ReactJs** for front-end of the web site and **Firebase** for authentication and verification of the email.

CODE FOR REGISTRATION AND LOGIN CREDENTIALS:

App.js

```
import './App.css';
import {BrowserRouter as Router, Switch, Route} from 'react-router-dom'
import {useState, useEffect} from 'react'
import {auth} from './firebase'
import {onAuthStateChanged} from 'firebase/auth'
import {AuthProvider} from './AuthContext'
import Profile from './Profile'
import Register from './Register'
import VerifyEmail from './VerifyEmail';
import PrivateRoute from './PrivateRoute'
import Login from './Login'

function App() {
  const [currentUser, setCurrentUser] = useState(null)
  const [timeActive, setTimeActive] = useState(false)
  useEffect(() => {
    onAuthStateChanged(auth, (user) => {
      setCurrentUser(user)
    })
  }, [])
  return (
    <Router>
    <AuthProvider value={{currentUser, timeActive, setTimeActive}}>
    <Switch>
    <PrivateRoute exact path="/" component={Profile} />
```

```

<Route exact path="/login" component={Login} />
<Route exact path="/register" component={Register} />
<Route exact path="/verify-email" component={VerifyEmail} />
</Switch>
</AuthProvider>
</Router>
);
}
export default App;

```

verifyEmail.js

```

import './verifyEmail.css'
import {useAuthValue} from './AuthContext'
import {useState, useEffect} from 'react'
import {auth} from './firebase'
import {sendEmailVerification} from 'firebase/auth'
import {useHistory} from 'react-router-dom'

function VerifyEmail() {
  const {currentUser} = useAuthValue()
  const [time, setTime] = useState(60)
  const {timeActive, setTimeActive} = useAuthValue()
  const [setButtonDisabled] = useState(false)
  const history = useHistory()
  useEffect(() => {
    const interval = setInterval(() => {
      currentUser?.reload()
        .then(() => {
          if(currentUser?.emailVerified){
            clearInterval(interval)
            history.push('/')
          }
        })
        .catch((err) => {
          alert(err.message)
        })
    }, 1000)
  }, [history, currentUser])
  const resendEmailVerification = () => {
    setButtonDisabled(true)
    sendEmailVerification(auth.currentUser)
  }

```

```

.then(() => {
  setButtonDisabled(false)
  setTimeActive(true)
}).catch((err) => {
  alert(err.message)
  setButtonDisabled(false)
})
}
return (
  <div className='center'>
    <div className='verifyEmail'>
      <span>{currentUser?.email}</span>
      <h1>Verify your Email Address</h1>
      <p>
        <strong>A Verification email has been sent to:</strong><br/>
        <span>{currentUser?.email}</span>
      </p>
      <span>Follow the instruction in the email to verify your account</span>
      <button
        onClick={resendEmailVerification}
        disabled={timeActive}
      >Resend Email {timeActive && time}</button>
    </div>
  </div>
)
}
export default VerifyEmail

```

Authentication Helper

```

import React, {useContext} from 'react'

const AuthContext = React.createContext()
export function AuthProvider({children, value}) {
  return (
    <AuthContext.Provider value={value}>
      {children}
    </AuthContext.Provider>
  )
}
export function useAuthValue(){
  return useContext(AuthContext)
}

```

Firestore Config

```
import { initializeApp } from "firebase/app"
import { getAuth } from "firebase/auth"

const firebaseConfig = {
  apiKey: "AIzaSyCtbFSEloGPuWx6meLRUS-nP6Hczdxyx8I",
  authDomain: "smart-waste-management-57367.firebaseio.com",
  projectId: "smart-waste-management-57367",
  storageBucket: "smart-waste-management-57367.appspot.com",
  messagingSenderId: "150430850891",
  appId: "1:150430850891:web:8a058ba0beef50dcb212ce",
  measurementId: "G-6BYLQYGJDG"
};

// Initialize Firebase and Firebase Authentication
const app = initializeApp(firebaseConfig)
const auth = getAuth(app)
export {auth}
```

login.js

```
import {useState} from 'react'
import { Link } from 'react-router-dom'
import './forms.css'
import {signInWithEmailAndPassword, sendEmailVerification } from
'firebase/auth'
import {auth} from './firebase'
import {useHistory} from 'react-router-dom'
import {useAuthValue} from './AuthContext'

function Login(){
  const [email, setEmail] = useState('')
  const [password, setPassword] = useState('')
  const [error, setError] = useState('')
  const {setTimeActive} = useAuthValue()
  const history = useHistory()
  const login = e => {
    e.preventDefault()
    signInWithEmailAndPassword(auth, email, password)
    .then(() => {
      if(!auth.currentUser.emailVerified) {
```

```

sendEmailVerification(auth.currentUser)
.then(() => {
  setTimeActive(true)
  history.push('/verify-email')
})
.catch(err => alert(err.message))
}else{
  history.push('/')
}
})
.catch(err => setError(err.message))
}
return(
<div className='center'>
<div className='auth'>
<h1>Log in</h1>
{error && <div className='auth__error'>{error}</div>}
<form onSubmit={login} name='login_form'>
<input
type='email'
value={email}
required
placeholder="Enter your email"
onChange={e => setEmail(e.target.value)}/>
<input
type='password'
value={password}
required
placeholder='Enter your password'
onChange={e => setPassword(e.target.value)}/>
<button type='submit'>Login</button>
</form>
<p>
Don't have an account?
<Link to='/register'>Create one here</Link>
</p>
</div>
</div>
)
}
export default Login

```

profile.js

```
import './profile.css'
import {useAuthValue} from './AuthContext'
import { signOut } from 'firebase/auth'
import { auth } from './firebase'

function Profile() {
  const {currentUser} = useAuthValue()
  return (
    <div className='center'>
      <div className='profile'>
        <h1>Profile</h1>
        <p><strong>Email: </strong>{currentUser?.email}</p>
        <p>
          <strong>Email verified: </strong>
          {`${currentUser?.emailVerified}`}
        </p>
        <span onClick={() => signOut(auth)}>Sign Out</span>
      </div>
    </div>
  )
}
export default Profile
```

verifyEmail.js

```
import './verifyEmail.css'
import {useAuthValue} from './AuthContext'
import {useState, useEffect} from 'react'
import {auth} from './firebase'
import {sendEmailVerification} from 'firebase/auth'
import {useHistory} from 'react-router-dom'

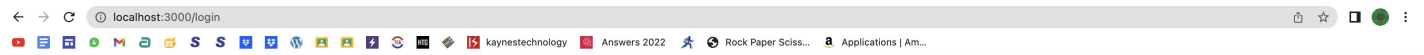
function VerifyEmail() {
  const {currentUser} = useAuthValue()
  const [time, setTime] = useState(60)
  const {timeActive, setTimeActive} = useAuthValue()
  const [setButtonDisabled] = useState(false)
  const history = useHistory()
  useEffect(() => {const interval = setInterval(() => {
```

```

currentUser?.reload()
.then(() => {
  if(currentUser?.emailVerified){
    clearInterval(interval)
    history.push('/')
  }
})
.catch((err) => {
  alert(err.message)
})
}, 1000)
}, [history, currentUser])
const resendEmailVerification = () => {
  setButtonDisabled(true)
  sendEmailVerification(auth.currentUser)
  .then(() => {
    setButtonDisabled(false)
    setTimeActive(true)
  }).catch((err) => {
    alert(err.message)
    setButtonDisabled(false)
  })
}
return (
  <div className='center'>
    <div className='verifyEmail'>
      <span>{currentUser?.email}</span>
      <h1>Verify your Email Address</h1>
      <p>
        <strong>A Verification email has been sent to:</strong><br/>
        <span>{currentUser?.email}</span>
      </p>
      <span>Follow the instruction in the email to verify your account</span>
      <button
        onClick={resendEmailVerification}
        disabled={timeActive}
      >Resend Email {timeActive && time}</button>
    </div>
  </div>)
}
export default VerifyEmail

```

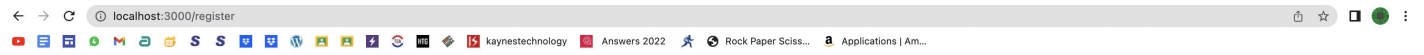
OUTPUTS



Log in

Login

Don't have an account?[Create one here](#)



Register

Register

Already have an account?[login](#)

vigneshsrinivasan772@gmail.com

Verify your Email Address

A Verification email has been sent to:
vigneshsrinivasan772@gmail.com

Follow the instruction in the email to verify your account

Resend Email 60

Your email has been verified

You can now sign in with your new account



1hx03x.internetofthings.ibmcloud.com/dashboard/devices/browse

IBM-EPBL/IBM-Pr... IBM-Project-1125... http://www.google... IBM

IBM Watson IoT Platform7376191ec309@smartinternz.comID: 1hx03x

Browse

Action

Device Types

Interfaces

Add Device

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location	Added By	Device Class
232323	Connected	cloud	Device	Nov 20, 2022 12:23 PM		7376191ec309@smartinternz.com	

Identity

Device Information

Recent Events

State

Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
status	{"Distance":74,"latitude":11.225130346002414...	json	a few seconds ago
status	{"Distance":86,"latitude":11.266726469416076...	json	a few seconds ago
status	{"Distance":56,"latitude":11.180323512389778...	json	a minute ago
status	{"Distance":94,"latitude":11.236415675597819...	json	a minute ago
status	{"Distance":72,"latitude":11.250703550292133...	json	a minute ago

> cloud_1 Disconnected cloud Device Nov 20, 2022 1:10 PM 7376191ec309@smartinternz.com

Items per page 50 | 1-2 of 2 Items1 of 1 page< 1 >

WATSON IOT RESULTS