

Project Report
On
USER AUTHENTICATION

**Submitted by A.Rishikesh,
J.Sai sindhu,V.Nandini**

**Under the guidance of
Ms Shalima sulthana Assistant Professor
Department of Computer Science and
Engineering**



**Rajiv Gandhi University of Knowledge and Technologies R.K.Valley,
Kadapa**

Andhrapradesh



Edit with WPS Office



Rajiv Gandhi University of Knowledge Technologies

RK Valley, Kadapa (Dist), Andhra Pradesh, 516330

DECLARATION

We hereby declare that the report of the B.Tech Mini Project Work entitled “**USER AUTENTICATION**” which is being submitted to Rajiv Gandhi University of Knowledge Technologies, RK Valley, in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide report of the work carried out by me. The material contained in this report has not been submitted to any university or institution for award of any degree.

A.Rishikesh – R170998

J.Sai sindhu-R170482

V.Nandini-R171100



**RAJIV GANDHI UNIVERSITY OF KNOWLEDGE
TECHNOLOGIES**



RGUKT

(A.P.Government Act 18 of 2008) RGUKT, RK VALLEY
Department of Computer Science and Engineering

CERTIFICATE FOR PROJECT COMPLETION

This is certify that the project entitled **“USER AUTHENTICATION”** submitted by **A.Rishikesh,J.Sai sindhu,V.Nandini(R170998,R170482, R171100)** under our guidance and supervision for the partial fulfillment for the degree Bachelor of Technology in Computer Science and Engineering during the academic year 2021-2022 at RGUKT, RK VALLEY. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any University or Institute for the award of any degree or diploma.

Project Internal Guide

Ms Shalima sulthana

Mr.P.

Harinadha

Head of the Department



Acknowledgement

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success. I am extremely grateful to our respected Director, Prof. K. SANDHYA RANI for fostering an excellent academic climate in our institution. I also express my sincere gratitude to our respected Head of the Department Mr P. HARINADHA for their encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project. I would like to convey thanks to our guide at college Ms. SHALIMA SULTHANA for their guidance, encouragement, co-operation and kindness during the entire duration of the course and academics. My sincere thanks to all the members who helped me directly and indirectly in the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement.



INDEX

S.NO	INDEX	PAGE NUMBER
1	Abstract	6
2	Introduction	7
3	Purpose	8
4	Scope	9
5	Advantages	9
6	Requirement Specification	10
7	Modules	11
8	Technologies Used	13
9	Use case Diagram, ERDiagram	15-16
10	Soure code, Output	17-30
11	conclusion	31
12	references	33

Abstrac

With the rapid evolution of the wireless communication technology, user authentication is important in order to ensure the security of the wireless communication technology.

Password play an important role in the process of authentication.

In the process of authentication, the password enter by the user will be transmitted along the traffic to the authentication server in order to allow the server to grant access to the authorised user. The attackers will use the chance to attempt to sniff others person password in order to perform some illegal activities by using other's identity to keep them safe from trouble. Due to the issues, there are many solutions has been proposed to improve the security of wireless communication technology. In this paper, the previously proposed solution will be used to enhance the security of the system.



INTRODUCTION

N

- Authentication is an activity to authenticate the person credential that wishes to perform the activity. In the process of authentication, the password enter by the user will be transmitted along the traffic to the authentication server in order to allow the server to grant access to the authorized user.
- When the password is transmitted, the attackers will try to sniff into the network to obtain data that include the user's password.
- Currently, there is rainbow table which able to trace the password with the hash algorithm to obtain the user's password.



Purpose

User authentication verifies the identity of a user attempting to gain access to a network or computing resource by authorizing a human-to-machine transfer of credentials during interactions on a network to confirm a user's authenticity.

Intended Audience

- 1)User
- 2)Server

product vision

The main vision of this project is to make authentication enables organizations to keep their networks secure by permitting only authenticated users or processes to gain access to their protected resources.



Scope

- During user authentication, a consent screen can be shown to end users, which can be useful for visualizing scopes and claims that will be issued by the authorization server.

Advantages:

- Provides greater security.
- Protects against brute force attacks.
- Reduces cost in the long run.
- User authentication ensures security without requiring such complicated policies.



Requirement Specification

Hardware

Configuration: Client

Ram	512 MB
Hard disk	10GB
Processor	1.0 GHz

Server side:

Ram	1GB
Hard disk	20GB
Processor	2.0GHz

Software

Requirements:	Html,Css,javascript
Server side Language	Nodejs,JWT
Database Server	MongoDB
Web Browser	Firefox,Chrome or any compatible Browser
Operating System	Ubuntu,Windows or any Compatible Browser
Software	EPASS



MODULES

- Server
- User
- Approval Process
- Status

Modules Description

□ **Server Module**

In server side module the details/information like User ID,Email,Phone No,Profession(Student/Teacher), Password.These are the details stored by the server module.

□ **User Module**

Each student /Teacher will have unique id for login into the system,login username and password is valid means he/she continue to view all the details and can update the previous register details also.This record will automatically send to the Server account. Server will view all the data and send the confirmation to the particular student/Teacher through application

□ Approval Process

This module is very important module in this project. The main aim of this module is to create the account of the user. This module is handling by the server and server only having all the rights to create the account. Server first of all view the entire details and view the each user details for creating new account. Suppose if the user enter the valid details the account will successfully created and the details of the user will be shown in the About you page.

□ Status

In this module the user can view the status of the account whether it is created successfully or not .



Technologies Used

- **FRONTEND** : HTML ,CSS , JavaScript
- **BACKEND** : MongoDB,Node.js,JWT.
- **ENVIRONMENT** :Visual Studio Code.

MongoDB

MongoDB is a source available cross-platform document-oriented database program. Classified as a NoSQL database program.MongoDB uses JSON-like documents with optional schemes.MongoDB is built on a scale-out architecture that has become popular with developers of all kinds for developing scalable applications.

JWT

JSON Web token is a proposed internet standard for creating data with optional signature or optional encryption.The tokens are signed either using a private secret or a public / private key. JSW is used to share security information between two parties a client and a server.



Node.js

Node.js is an open source, cross platform, backend javascript runtime environment that runs on a javascript engine and executes javascript code outside a web server. Node.js is primarily used for non-blocking, event-driven servers due to its single threaded nature.

HTML

HTML, or Hypertext Markup Language, is a markup language for the web that defines the structure of web pages.

Hypertext: text (often with embeds such as images, too) that is organized in order to connect related items

Markup: a style guide for typesetting anything to be printed in hardcopy or soft copy format

Language: a language that a computer system understands and uses to interpret commands.

CSS

Cascading Style Sheets (CSS) is a Style sheet Language used for describing the Presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World wide web, alongside HTML and Javascript(js).

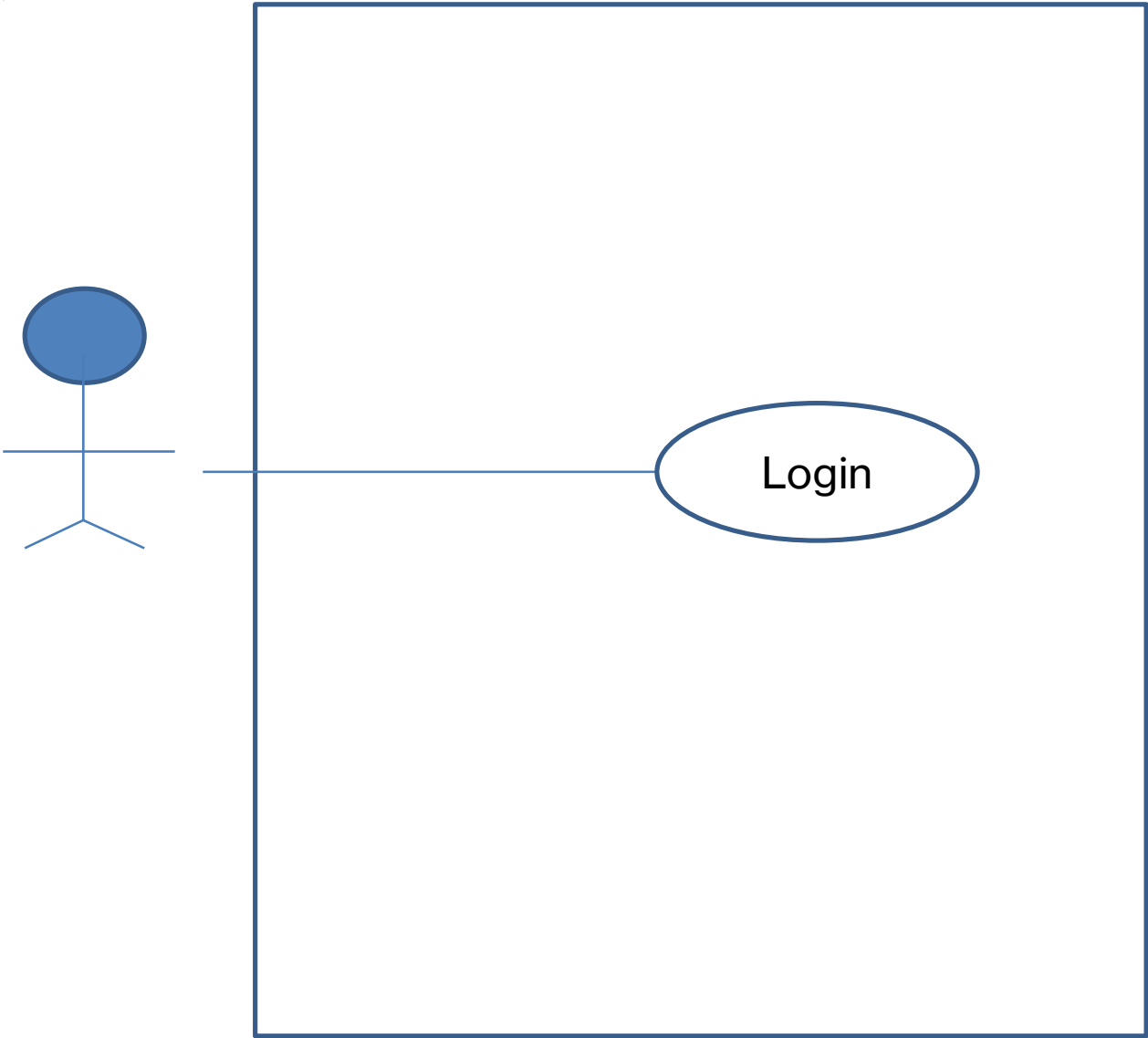
Javascript

JavaScript is a dynamic programming language that's used for web development, in web applications, for game development,

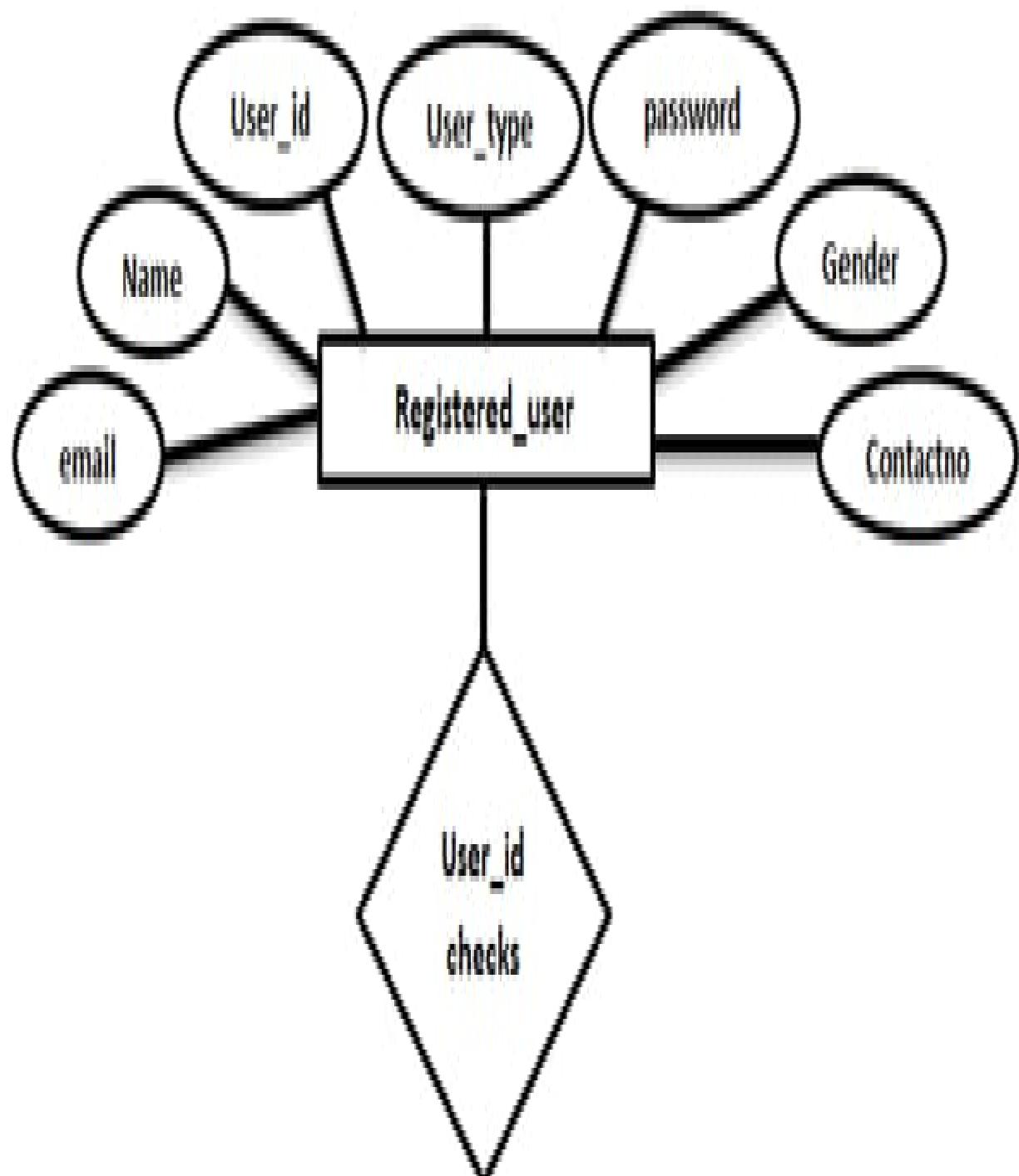


Use case Diagram

Secure User Authentication System



ER Diagram



SOURCE

```
{ } package-lock.json  < index.html  JS App.js  X

Frontend > src > JS App.js > [x] App

1  import React, {createContext,useReducer} from 'react';
2  import {initialState,reducer} from './reducer/UseReducer';
3  import Navbar from './components/Navbar';
4  import About from './components/About';
5  import Signup from './components/Signup';
6  import Login from './components/Login';
7  import Home from './components/Home';
8  import 'bootstrap/dist/css/bootstrap.css';
9  import {BrowserRouter as Router,Switch,Route} from "react-router-dom";
10 import Errorpage from './components/Errorpage';
11 import Logout from './components/Logout';
12
13 export const userContext = createContext();
14 // 1.context api
15 const Routing = () =>{
16   return (<Router>
17     <Navbar />
18     <div className="container mt-3">
19       <Switch>
20         <Route exact path="/">
21           <Home />
22         </Route>
23         <Route exact path="/about">
24           <About />
25         </Route>
26         <Route exact path="/signup">
27           <Signup />
28         </Route>
29         <Route exact path="/login">
30           <Login />
31         </Route>
32
```



Frontend > src > JS App.js > [App]

```
31     </Route>
32
33
34     <Route exact path="/logout">
35       <Logout />
36     </Route>
37
38     <Route >
39       <Errorpage />
40     </Route>
41
42   </Switch>
43 </div>
44 </Router>)
45 }
46
47 const App = () =>{
48
49
50   const [state,dispatch] = useReducer(reducer,initialState);
51
52
53   return (
54     <>
55       <userContext.Provider value={{state,dispatch}}>
56         <Routing />
57       </userContext.Provider>
58     </>
59   );
60 }
61
62 export default App;
```



OUTPUT



Login

Sign up

About You

HOLA

Please log in

digital specialist role



Edit with WPS Office

package-lock.json

index.html

Frontend > public > index.html > html > head > link

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
6     <meta name="viewport" content="width=device-width, initial-scale=1" />
7     <meta name="theme-color" content="#000000" />
8     <meta
9       name="description"
10      content="Web site created using create-react-app"
11    />
12    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
13    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
14      integrity="sha384-1BmE4kWBq78iYhFtdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
15    <!--
16      manifest.json provides metadata used when your web app is installed on a
17      user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/
18    -->
19    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
20    <!--
21      Notice the use of %PUBLIC_URL% in the tags above.
22      It will be replaced with the URL of the 'public' folder during the build.
23      Only files inside the 'public' folder can be referenced from the HTML.
24
25      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
26      work correctly both with client-side routing and a non-root public URL.
27      Learn how to configure a non-root public URL by running 'npm run build'.
28    -->
29    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/material-design-iconic-font/2.2.0/
30      css/material-design-iconic-
31      font.min.css">
32    <title>React App</title>
33  </head>
```



Frontend > public > index.html > ...

```
18  -->
19  <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
20  <!--
21  Notice the use of %PUBLIC_URL% in the tags above.
22  It will be replaced with the URL of the `public` folder during the build.
23  Only files inside the `public` folder can be referenced from the HTML.
24
25  Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
26  work correctly both with client-side routing and a non-root public URL.
27  Learn how to configure a non-root public URL by running `npm run build`.
28  --><link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/material-design-iconic-font/2.2.0/
29  css/material-design-iconic-
30  font.min.css">
31  <title>React App</title>
32  </head>
33  <body>
34  <noscript>You need to enable JavaScript to run this app.</noscript>
35  <div id="root"></div>
36
37  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
38  integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IIRH9sENB00LRn5q+8nbTov4+1p" crossorigin="anonymous">
39  </script>
40
41  </body>
42  </html>
43  |
```

OUTPUT

Sign up

 Username

sindhu

 Email

saisindhujupalli@gmail.com

 Phone Number

93475395799

 Profession

student

 Password

 Confirm Password

12345678

Signup

Login

localhost:3000 says

SUCCESS

OK



Backend > middleware > JS Authenticate.js > [0] Authenticate

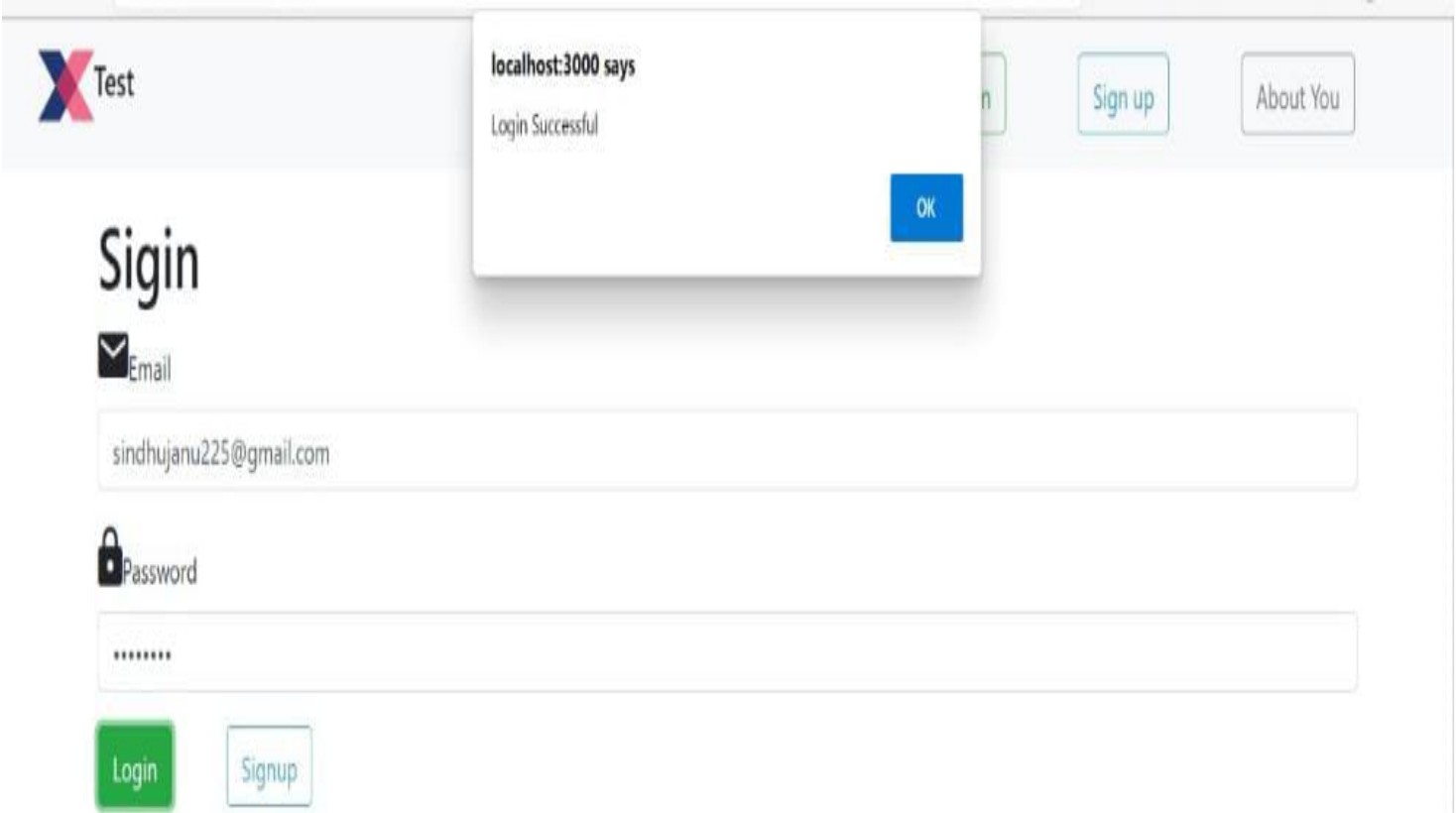
```

1  const jwt = require("jsonwebtoken");
2  const User = require("../model/userSchema");
3
4  const Authenticate = async (req, res, next) => {
5      try {
6          const token = req.cookies.jwtToken;
7          const verifyToken = jwt.verify(token, process.env.SECRET_KEY);
8
9          const rootUser = await User.findOne({ _id: verifyToken._id, "tokens.token": token });
10         if (!rootUser) {
11             throw new Error("User not found");
12         }
13         req.token = token;
14         req.rootUser = rootUser;
15         req.rootUserId = rootUser._id;
16
17         next();
18
19     } catch (err) {
20         res.status(401).send("Unauthorized");
21         console.log(err);
22     }
23
24 }
25
26
27 }
28 module.exports = Authenticate

```



OUTPUT



sindhu



User Id	6329e382bcf595022e6b7fa4
Name	sindhu
Email	sindhujanu225@gmail.com
Profession	student
Phone Number	9478287482

```
Backend > JS app.js > ...
1  const express = require("express");
2  const dotenv = require("dotenv");
3  const mongoose = require("mongoose");
4  const cookieParser = require("cookie-parser");
5  const app = express();
6
7
8
9
10 // db connection
11 // const DB = "mongodb+srv://love:love@cluster0.vhpdn.mongodb.net/myFirstDatabase?retryWrites=true&w=maj
12 // app.use(express.json());
13 dotenv.config({path:"./.env"});
14
15 require('./db/conn');
16 app.use(express.json());
17 app.use(cookieParser());
18 // const User = require("./model/userSchema");
19 app.use(require('./router/auth'));
20
21 const PORT = process.env.PORT || 5000;
22
23
24 // Middleware
25 // const middleware = (req,res,next)=>{
26 //     console.log(`middleware`);
27 //     next();
28 // }
```



{ package-lock.json

<> index.html ●

JS index.js

JS app.js

×

Backend > JS app.js > ...

```
30
31 // app.get('/',(req,res)=>{
32 //     res.send(`Hello backend`);
33 // });
34
35 // app.get('/about',middleware,(req,res)=>{
36 //     console.log(`about`);
37 //     res.send(`Hello about`);
38 // });
39
40 // app.get('/contact',(req,res)=>{
41 //     // res.cookie("name","love");
42 //     res.send(`Hello contact`);
43 // });
44
45 app.get('/signin',(req,res)=>{
46 |     res.send(`Hello signin`);
47 | });
48
49 app.get('/signup',(req,res)=>{
50 |     res.send(`Hello signup`);
51 | });
52
53 app.listen(PORT,()=>{
54 |     console.log(`Server is running on port ${PORT}`);
55 | }
56 | );
```



OUTPUT

[Login](#)[Sign up](#)[About You](#)

HOLA

sindhu

Welcome to licet portal

your infosys exam has been scheduled on 28/12/2022



Edit with WPS Office

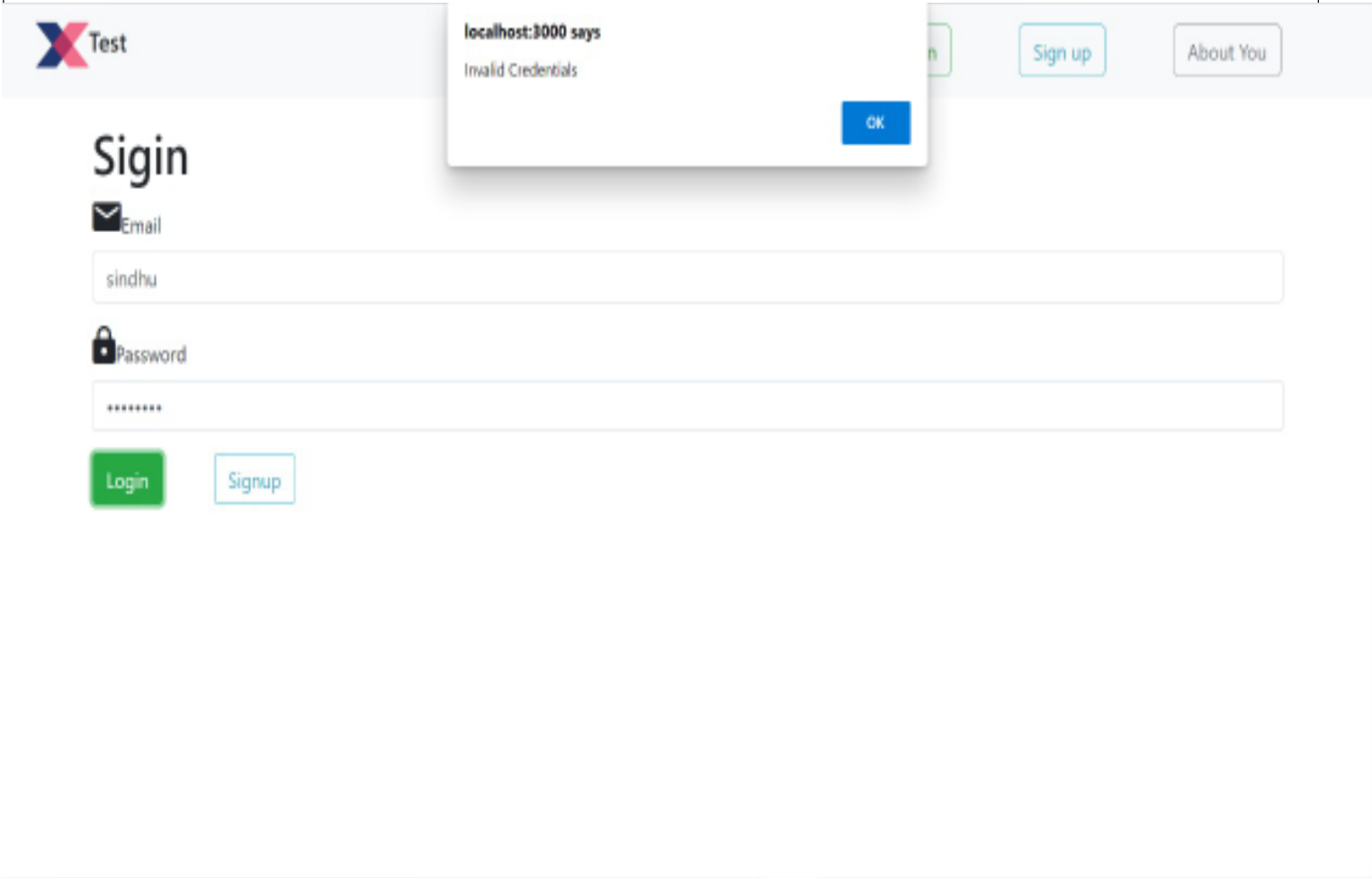
28 page of
33

Backend > {} package.json > ...

```
1  {
2    "name": "agneslily",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1",
8      "start": "nodemon app.js"
9    },
10   "keywords": [],
11   "author": "",
12   "license": "ISC",
13   "dependencies": {
14     "bcryptjs": "^2.4.3",
15     "cookie-parser": "^1.4.6",
16     "dotenv": "^16.0.0",
17     "emailjs": "^4.0.0",
18     "express": "^4.18.0",
19     "jsonwebtoken": "^8.5.1",
20     "mongoose": "^6.3.1",
21     "multer": "^1.4.4",
22     "nodemon": "^2.0.15"
23   }
24 }
25
```



OUTPUT



Conclusio

n

This web application provides authentication for an online test portal. It saves time as it allows the students in the University/in the outside to use their unique details to sign up with xtest online test portal so users later can use their credentials to login into the System, being web-based, it needs to be thoroughly tested before implementation to find any security gaps.



Future Enhancement

- The future of authentication does not include more complex passwords or passphrases or better user authentication. But it would be easy for systems to generate passwords with increased security and need will disappear for users to remember the password.
- Authentication likely sees a surge of evolution in the background; such developments that are invisible to the user. These developments are



References

□ <https://www.w3schools.com>

□ <https://www.arangodb.com/>

□ <https://www.javatpoint.com/>

□ <https://www.geeksforgeeks.org/>

