**1.** **Develop a login module using express middleware. When proper credentials are given display a greeting with an image and happy birthday text. This page must have a logout link. When you logout get redirected to login page. If credentials are incorrect, then redirect to login page. After login create session and display the session id also. Destroy the session on logout and set session timeout as 5 hours.**

**CODE:**

const express = require('express');

const session = require('express-session');

const bodyParser = require('body-parser');

const app = express();

app.use(bodyParser.urlencoded({ extended: true }));

app.use(session({

secret: 'mySecretKey',

resave: false,

saveUninitialized: true,

cookie: { maxAge: 5 \* 60 \* 60 \* 1000 }

}));

const isLoggedIn = (req, res, next) => {

if (req.session && req.session.user) {

next();

} else {

res.redirect('/login');

}

};

app.get('/login', (req, res) => {

if (req.session && req.session.user) {

res.redirect('/dashboard');

} else {

res.send(`

<h1>Login Page</h1>

<form method="post" action="/login">

Username: <input type="text" name="username" required><br>

Password: <input type="password" name="password" required><br>

<button type="submit">Login</button>

</form>

`);

}

});

const validUsername = 'user';

const validPassword = 'password';

app.post('/login', (req, res) => {

const { username, password } = req.body;

if (username === validUsername && password === validPassword) {

req.session.user = username;

res.redirect('/dashboard');

} else {

res.redirect('/login');

}

});

app.get('/dashboard', isLoggedIn, (req, res) => {

res.send(`

<h1>Welcome ${req.session.user}!</h1>

<img src="https://placekitten.com/200/300" alt="Kitten">

<p>Happy Birthday!</p>

<a href="/logout">Logout</a>

<p>Session ID: ${req.sessionID}</p>

`);

});

app.get('/logout', (req, res) => {

req.session.destroy((err) => {

if (err) {

console.error(err);

}

res.redirect('/login');

});

});

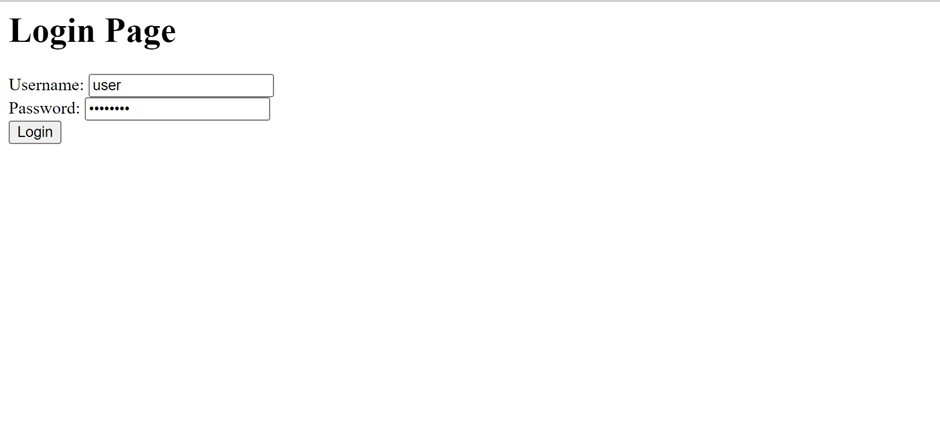
const PORT = 3000;

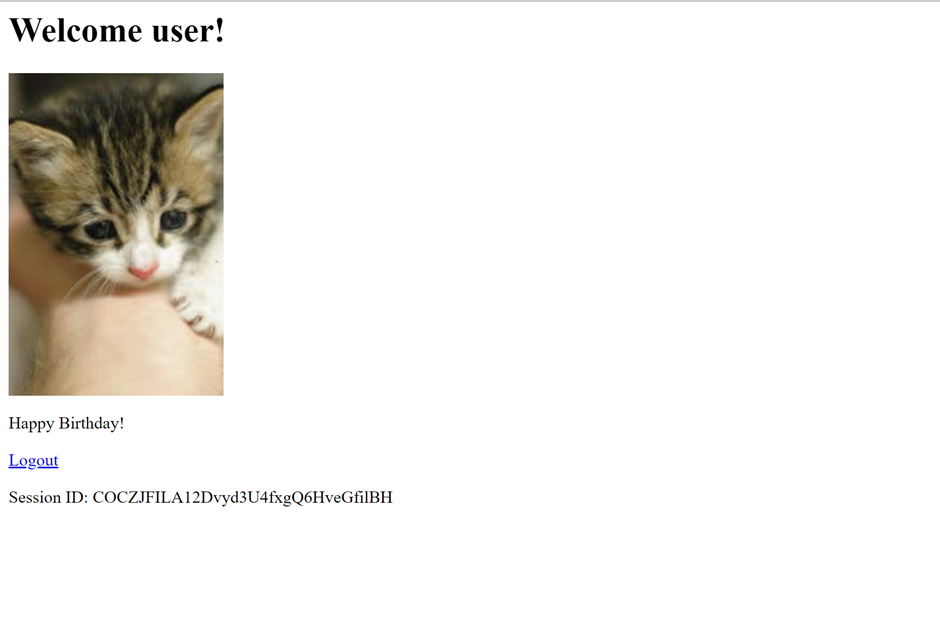
app.listen(PORT, () => {

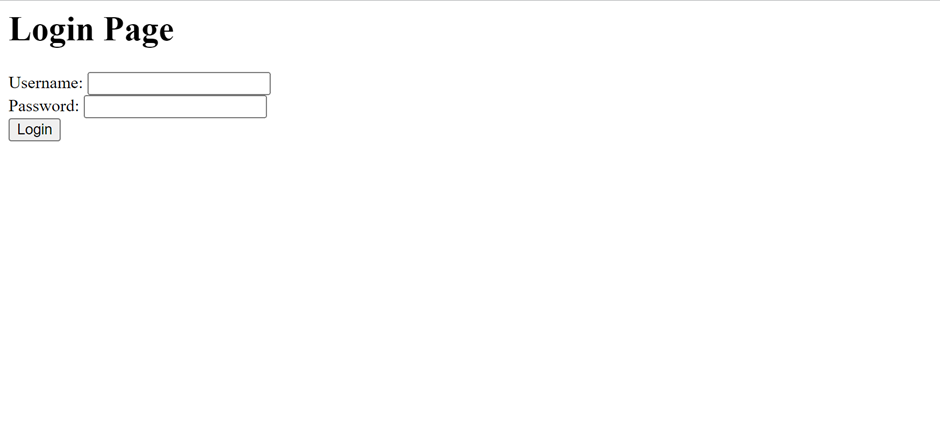
console.log(`Server running on port ${PORT}`);

});

**OUTPUT:**

****

****

****

**2.** **Create a stateful class component using react called patient. The component must render 3 props namely patient name, sugar in fasting and after food in <h1> tag. The text must be "name props has fasting props and after food props". Create 2 such patient tags. Below each patient tag add a horizontal rule.**

**CODE:**

import React, { Component } from 'react';

class Patient extends Component {

render() {

const { name, fasting, afterFood } = this.props;

return (

<div>

<h1>{`${name} has fasting ${fasting} and after food ${afterFood}`}</h1>

<hr />

</div>

);

}

}

class App extends Component {

render() {

return (

<div>

<Patient name="John" fasting="90" afterFood="120" />

<Patient name="Alice" fasting="95" afterFood="110" />

</div>

);

}

}

export default App;

**OUTPUT:**



**3.** **Create a mongodb collection for a pharmacy. Insert a record of medicine name and cost. Delete a record with medicine name "crocin". Update the cost of Asthalin to 150. Display the names of all the medicines that start with S.**

**CODE:**

Create collection:

db.medicines.insertMany([

{ name: "Crocin", cost: 50 },

{ name: "Asthalin", cost: 100 },

{ name: "Saridon", cost: 80 },

{ name: "Spasmo-Proxyvon", cost: 120 }

])

Delete crocin:

db.medicines.deleteOne({ name: "Crocin" })

Update the Cost of Asthalin to 150:

db.medicines.updateOne(

{ name: "Asthalin" },

{ $set: { cost: 150 } }

)

Display Medicines Starting with "S":

db.medicines.find({ name: /^S/ })

**OUTPUT:**





