7. Develop an authentication mechanism with email_id and password using HTML and Express JS (POST method)

```
mkdir PGM7_RE

cd PGM_RE

npm init -y

npm install express body-parser bcrypt
```

server.js /// create file inside PGM7_RE

```
const express = require('express');

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

const bcrypt = require('bcrypt');

const path = require('path');


const app = express();

const PORT = 3000;


// Middleware to parse URL-encoded and JSON request bodies

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

app.use(bodyParser.json());


// Serve static files (your HTML)

app.use(express.static(path.join(__dirname, 'public')));


// In-memory user storage (for demonstration purposes only - use a database in a real application)

const users = [];


// --- Registration Route ---
```

```
app.post('/register', async (req, res) => {
  const { email, password } = req.body;

  // Check if the email already exists
  if (users.find(user => user.email === email)) {
    return res.status(400).send('Email already registered.');
  }

  try {
    // Hash the password
    const hashedPassword = await bcrypt.hash(password, 10);

    // Store the new user (in a real app, you'd save to a database)
    users.push({ email, password: hashedPassword });
    res.status(201).send('Registration successful!');
  } catch (error) {
    console.error('Error during registration:', error);
    res.status(500).send('Registration failed.');
  }
});

// --- Login Route ---
app.post('/login', async (req, res) => {
  const { email, password } = req.body;

  // Find the user by email
  const user = users.find(user => user.email === email);

  if (!user) {
    return res.status(401).send('Invalid credentials.');
  }

  try {
    // Compare the provided password with the stored hashed password
    const passwordMatch = await bcrypt.compare(password, user.password);

    if (passwordMatch) {
      res.status(200).send('Login successful!');
      // In a real application, you would typically generate and send a session token or
cookie here.
    } else {
      res.status(401).send('Invalid credentials.');
    }
  } catch (error) {
    console.error('Error during login:', error);
    res.status(500).send('Login failed.');
  }
});

app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});
```

Create a public folder in your project directory. Inside public, create two HTML files: register.html and login.html.

public/register.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Register</title>

  <style>

    body { font-family: sans-serif; display: flex; justify-content: center; align-items:
center; min-height: 100vh; background-color: #f4f4f4; }

    .container { background-color: #fff; padding: 30px; border-radius: 8px; box-
shadow: 0 2px 4px rgba(0, 0, 0, 0.1); }

    h2 { text-align: center; margin-bottom: 20px; color: #333; }

    .form-group { margin-bottom: 15px; }

    label { display: block; margin-bottom: 5px; color: #555; }

    input[type="email"], input[type="password"] { width: 100%; padding: 10px;
border: 1px solid #ddd; border-radius: 4px; box-sizing: border-box; }

    button { background-color: #007bff; color: white; padding: 10px 15px; border:
none; border-radius: 4px; cursor: pointer; font-size: 16px; }

    button:hover { background-color: #0056b3; }

    .message { margin-top: 10px; font-size: 0.9em; }

    .success { color: green; }

    .error { color: red; }

  </style>

</head>

<body>

  <div class="container">

    <h2>Register</h2>

    <form id="registerForm">

      <div class="form-group">

        <label for="email">Email:</label>
```

```
          <input type="email" id="email" name="email" required>
        </div>
        <div class="form-group">
          <label for="password">Password:</label>
          <input type="password" id="password" name="password" required>
        </div>
        <button type="submit">Register</button>
        <div id="registerMessage" class="message"></div>
      </form>
      <p style="text-align: center; margin-top: 15px;">Already have an account? <a
href="/login.html">Login</a></p>
    </div>
    <script>
      document.getElementById('registerForm').addEventListener('submit', async
(event) => {
        event.preventDefault();
        const email = document.getElementById('email').value;
        const password = document.getElementById('password').value;
        const messageDiv = document.getElementById('registerMessage');

        try {
          const response = await fetch('/register', {
            method: 'POST',
            headers: {
              'Content-Type': 'application/json',
            },
            body: JSON.stringify({ email, password }),
          });

          const data = await response.text();
          if (response.ok) {
            messageDiv.className = 'message success';
            messageDiv.textContent = data;
            document.getElementById('registerForm').reset();
          } else {
            messageDiv.className = 'message error';
            messageDiv.textContent = data;
          }
        } catch (error) {
          console.error('Error during registration:', error);
          messageDiv.className = 'message error';
          messageDiv.textContent = 'An error occurred during registration.';
        }
      });
  </script>
</body>
</html>
```

public/login.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login</title>
  <style>
    body { font-family: sans-serif; display: flex; justify-content: center; align-items:
center; min-height: 100vh; background-color: #f4f4f4; }
    .container { background-color: #fff; padding: 30px; border-radius: 8px; box-
shadow: 0 2px 4px rgba(0, 0, 0, 0.1); }
```

```
        h2 { text-align: center; margin-bottom: 20px; color: #333; }
        .form-group { margin-bottom: 15px; }
        label { display: block; margin-bottom: 5px; color: #555; }
        input[type="email"], input[type="password"] { width: 100%; padding: 10px;
border: 1px solid #ddd; border-radius: 4px; box-sizing: border-box; }
        button { background-color: #007bff; color: white; padding: 10px 15px; border:
none; border-radius: 4px; cursor: pointer; font-size: 16px; }
        button:hover { background-color: #0056b3; }
        .message { margin-top: 10px; font-size: 0.9em; }
        .success { color: green; }
        .error { color: red; }
    </style>
</head>
<body>
    <div class="container">
        <h2>Login</h2>
        <form id="loginForm">
            <div class="form-group">
                <label for="email">Email:</label>
                <input type="email" id="email" name="email" required>
            </div>
            <div class="form-group">
                <label for="password">Password:</label>
                <input type="password" id="password" name="password" required>
            </div>
            <button type="submit">Login</button>
            <div id="loginMessage" class="message"></div>
        </form>
        <p style="text-align: center; margin-top: 15px;">Don't have an account? <a
href="/register.html">Register</a></p>
    </div>
```

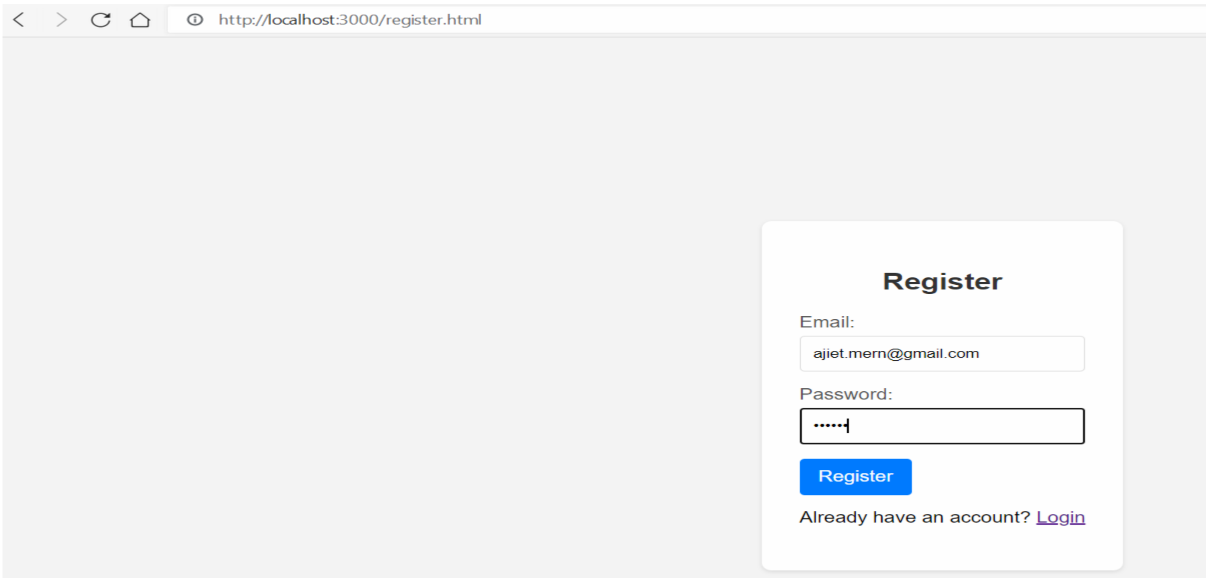
```
  <script>
    document.getElementById('loginForm').addEventListener('submit', async (event)
=> {
      event.preventDefault();
      const email = document.getElementById('email').value;
      const password = document.getElementById('password').value;
      const messageDiv = document.getElementById('loginMessage');

      try {
        const response = await fetch('/login', {
          method: 'POST',
          headers: {
            'Content-Type': 'application/json',
          },
          body: JSON.stringify({ email, password }),
        });

        const data = await response.text();

        if (response.ok) {
          messageDiv.className = 'message success';
          messageDiv.textContent = data;
          // In a real application, you would redirect the user to a dashboard or
protected route here.
          console.log('Login successful!');
        } else {
          messageDiv.className = 'message error';
          messageDiv.textContent = data;
        }
        } catch (error) {
            console.error('Error during login:', error);
            messageDiv.className = 'message error';
            messageDiv.textContent = 'An error occurred during login.';
        }
      });
  </script>
</body>
</html>
```

TO RUN:

node server.js

Open your web browser and go to http://localhost:3000/register.html or http://localhost:3000/login.html

<u>OUTPUT:</u>



http://localhost:3000/register.html

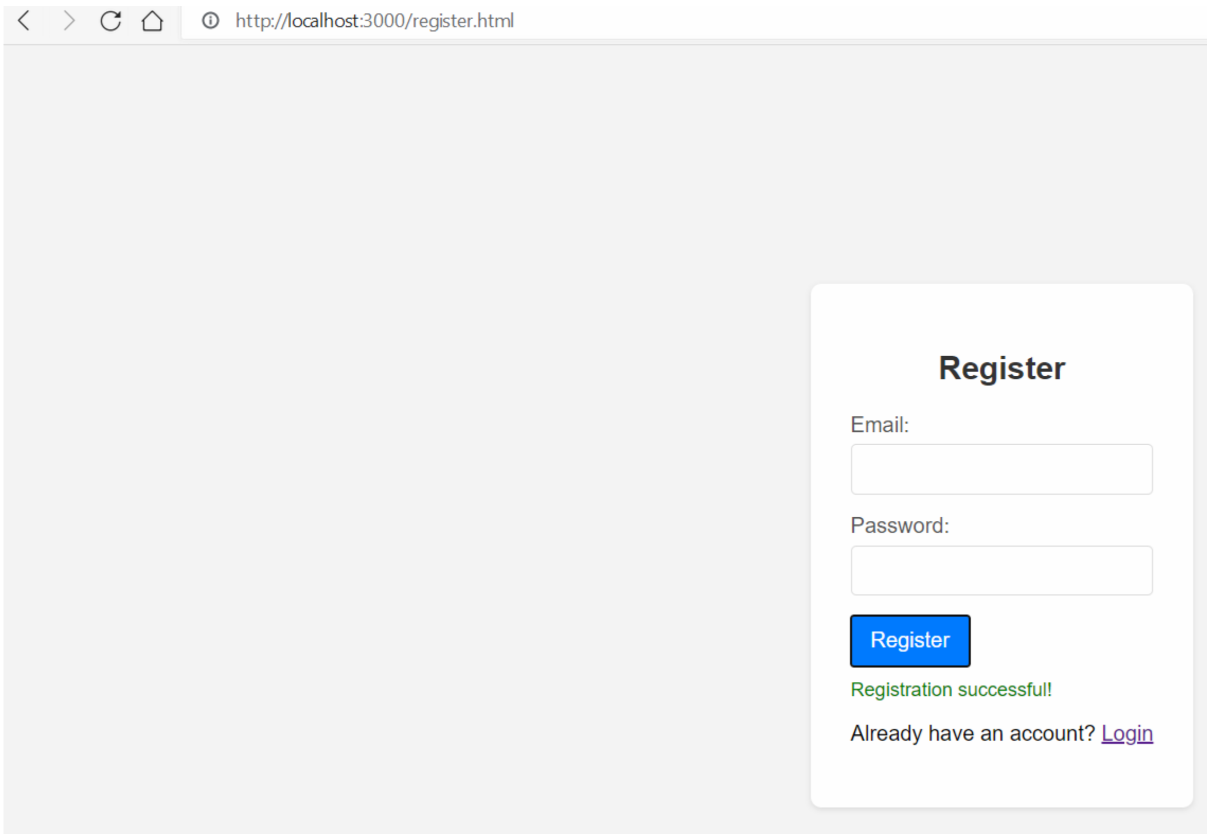
Register

Email:

ajiet.mern@gmail.com

Password:

••••••



http://localhost:3000/register.html

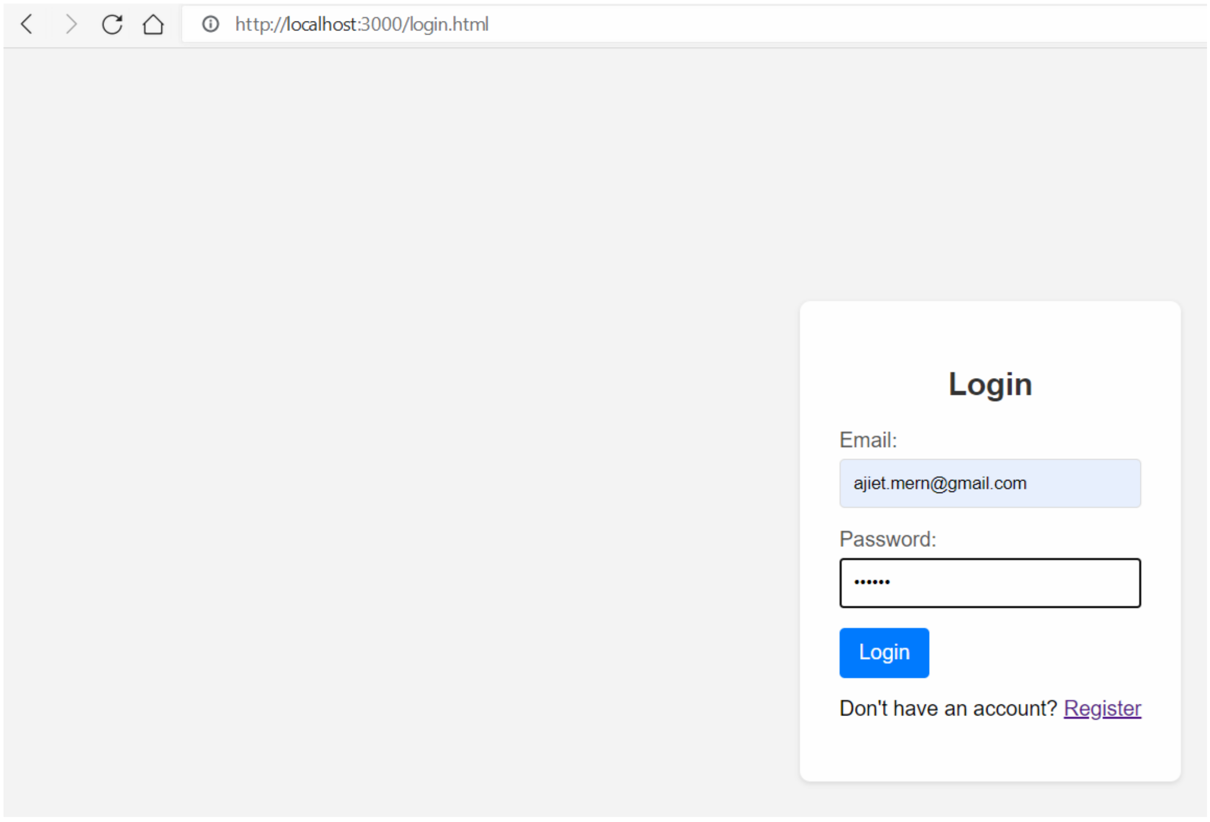
Register

Email:

Password:

Registration successful!

Already have an account? Login



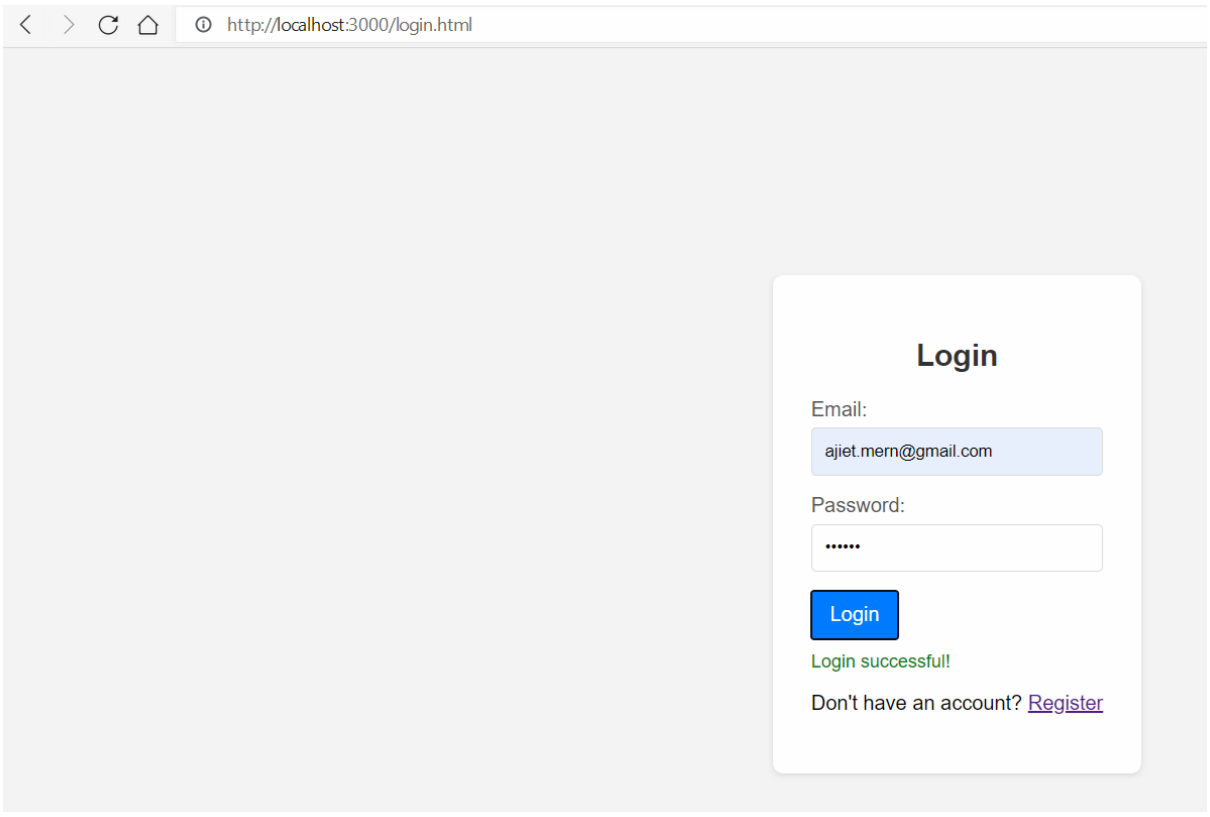
http://localhost:3000/login.html

Login

Email: ajiet.mern@gmail.com

Password: ••••••

Don't have an account? Register



http://localhost:3000/login.html

Login

Email:

ajiet.mern@gmail.com

Password:

••••••

Login

Login successful!

Don't have an account? Register