UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+ ACCREDITED UNIVERSITY

BACK END TECHNOLOGIES

Worksheet 3

Name:Prakhar Chauhan **UID:22MCA20056**
Branch: MCA **Section/Group:1/A**
Semester: Third **Date of Performance:10-09-2023**
Subject Code:22CAH-706

Aim Of The Practical - The aim of this practical is to create a user signup page, collect user information, store it in a MongoDB database, and document the process.

Task to be done:

- Design a signup page with HTML and CSS.
- Create a server using Node.js and Express.js to handle HTTP requests.
- Implement the MongoDB connection to store user data.
- Create API endpoints for user signup and handle form submissions.
- Document the process with a flowchart.
- Define learning outcomes.

Flowchart

Start
|
User visits signup page
|
User fills out the signup form
|
Form data submitted to the server
|
Server validates user input
|
If valid:
|
Store user data in MongoDB
|
Send success response to user
|
Redirect user to a welcome page
|
End

If invalid:
|
Send error response to user
|
User corrects the form
|
Loop back to the signup page

Code for experiment/practical:

Signup

```
<div className="container text-center text-dark mt-5">
  <div className="row">
    <div className="col-lg-6 d-block mx-auto mt-5">
      <div className="row">
        <div className="col-xl- col-md-12 col-md-12">
          <div className="wow-bg" id="formBg">
            <h3 className="colorboard">Signup</h3>
            <p className="text-muted">Create your UseCoupon account</p>
            <form>
              <div className="input-group mb-3">
                <input type="text" name="fname" id="" className="form-control textbox-dg" onChange={handleChange} placeholder='Enter Your Name' />
              </div>
              <div className="input-group mb-3">
                <input type="email" name="email" onChange={handleChange} className="form-control textbox-dg" placeholder="Email" />
              </div>
              <div className="input-group mb-4">
                <input type={!passhow ? "password" : "text"} name="password" id="" className="form-control textbox-dg" onChange={handleChange} placeholder='Enter Your password' />
                <div className='showpass' onClick={() => setPassShow(!passhow)} >
                  {!passhow ? "Show" : "Hide"}
                </div>
              </div>
              <div className="row">
                <div className="col-12">
                  <button type="submit" className="btn btn-primary btn-block logn-btn" onClick={handleSubmit}>Signup</button>
                  <p>You have Already Account? <NavLink to="/">Login</NavLink> </p>
                </div>
              </div>
            </form>

          </div>
        </div>
      </div>
    </div>
```

User Schema

```
const userSchema = new mongoose.Schema({
    fname:{
        type: String,
        required: true,
        trim: true
    },
    email:{
        type: String,
        required: true,
        unique: true,
        validate(value){
            if(!validator.isEmail(value)){
                throw new Error("Not Valid Email")
            }
        }
    },
    password:{
        type: String,
        required: true,
        minlength: 5
    },
    tokens:[
        {
            token:{
                type: String,
                required: true,
            }
        }
    ]
});
```

Login

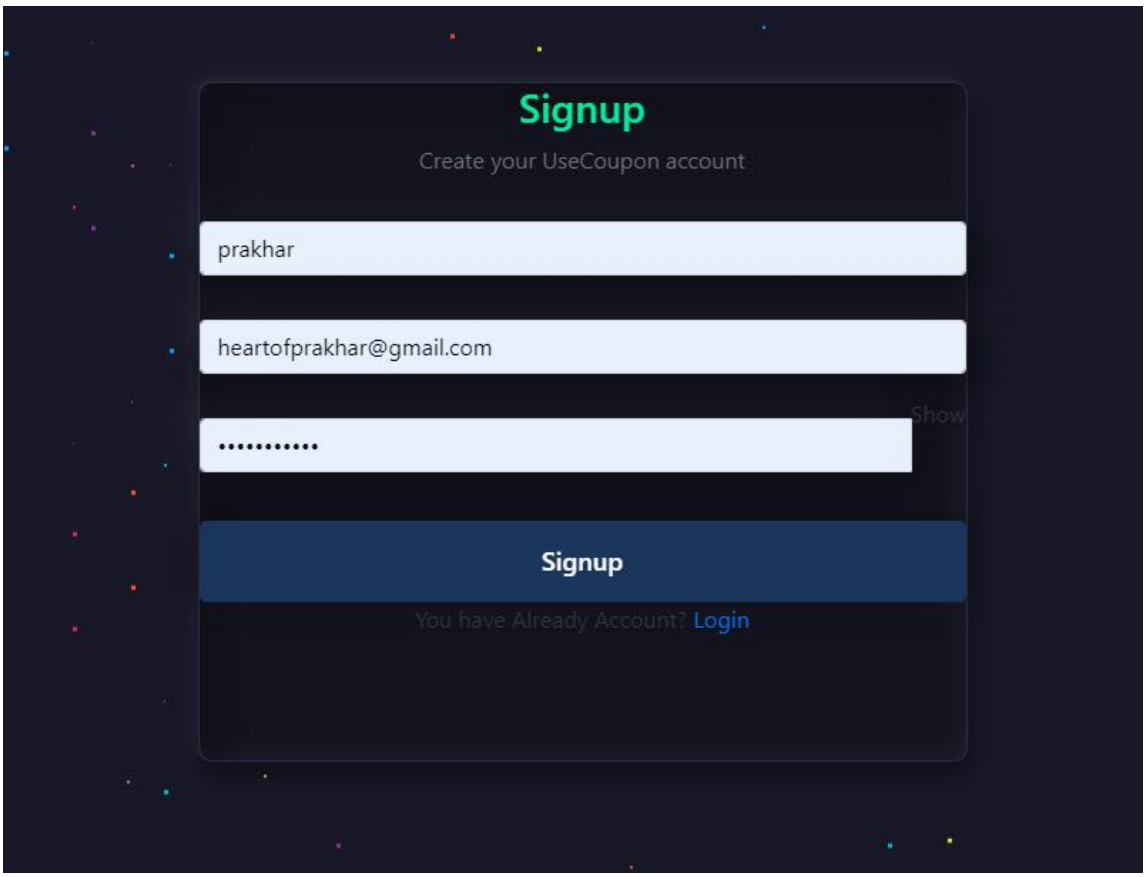
```
<div className="container text-center text-dark mt-5">
        <div className="row">
          <div className="col-lg-6 d-block mx-auto mt-5">
            <div className="row">
              <div className="col-xl- col-md-12 col-md-12">
                <div className="wow-bg" id="formBg">
              <h3 className="colorboard">Login</h3>
              <p className="text-muted">Sign In to your account</p>
              <form>

                    <div className="input-group mb-3">
                      <input type="email" name="email" onChange={(e) => setEmail(e.target.value)} className="form-control textbox-dg" placeholder="Email" />
                    </div>


                  </form>

              <div className="row">
                <div className="col-12">
                  {" "}
                  <button
                    type="button"
                    className="btn btn-primary btn-block logn-btn"
                    onClick={sendOtp}>Login
                    {
                        spiner ? <span><Spinner animation="border" />></span>:""
                    }


                  </button>{" "}
                  <p>You have not any Account? <NavLink to="/register">Register</NavLink> </p>
                </div>

              </div>
              <div className="mt-6 btn-list">
                <button
                  type="button"
                  className="socila-btn btn btn-icon btn-facebook fb-color"
                >
```

Database

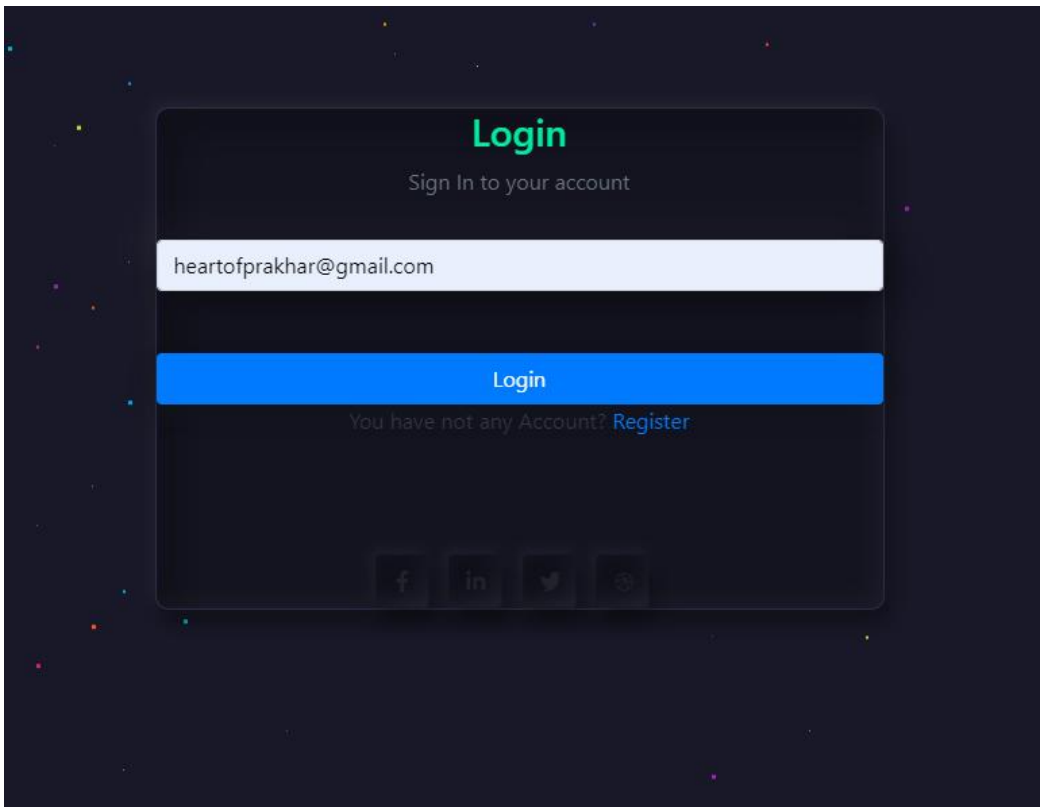
```
JS db.js ×
server > JS db.js > ...
1  const mongoose = require("mongoose");
2  const mongoURI =
3    "mongodb://localhost:27017/newcoupon";
4
5  const connectDB = () => {
6    mongoose.connect(mongoURI, () => {
7      console.log("connected to mongo Sucessfully");
8    });
9  };
10
11 module.exports = connectDB;
```

Output

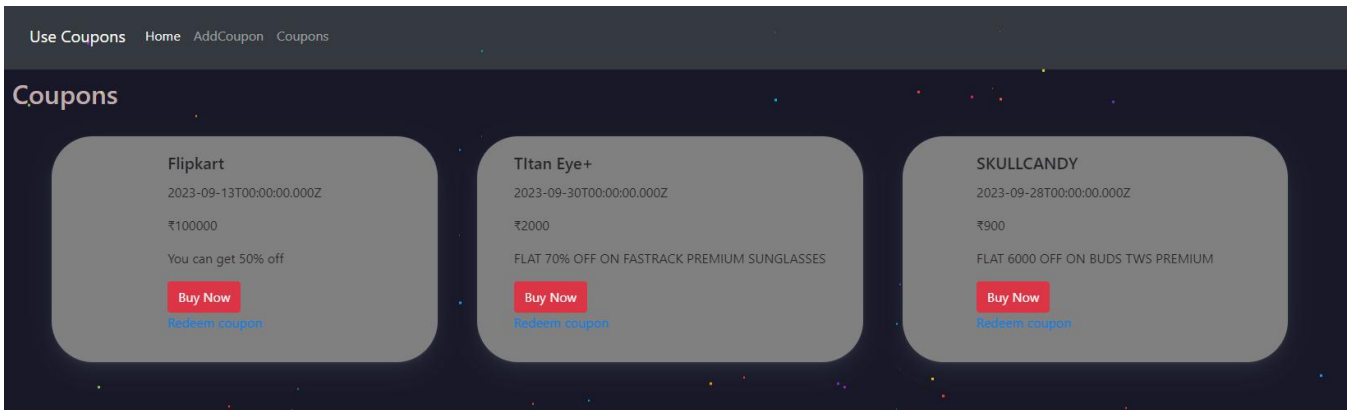
SIGNUP



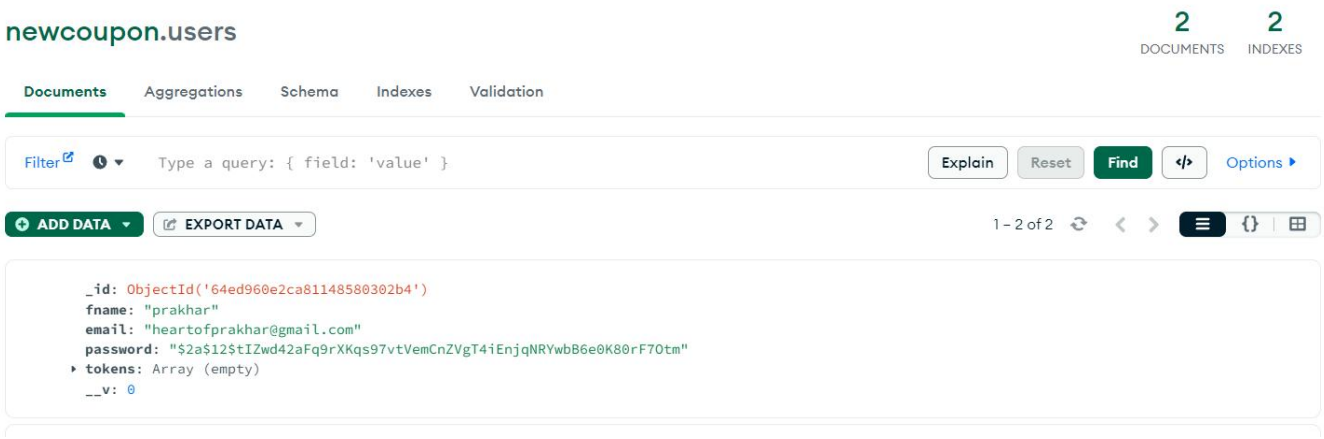
LOGIN



LOGGED IN TO HOME PAGE



SAVED USER INFO



Learning outcomes (What I have learnt):

- Create a web page with HTML and CSS for user interaction.
- Set up a server using Node.js and Express.js.
- Establish a connection to a MongoDB database.
- Create API endpoints to handle user data.
- Validate and store user data securely.
- Handle errors and provide appropriate responses to users.
- Document the entire process for reference.