# Deploy the Laptop Shopping Cart API on Netlify.

**Prepare Your API Code:**

Make sure your API code is ready and functional. It should be in a format that can be deployed as a serverless function on Netlify.

# Setup Your Code Repository:

If your API code is not already in a code repository, create a Git repository for it (e.g., using GitHub, GitLab, or Bitbucket). Make sure your code is committed and pushed to the repository.

# Install Netlify CLI:

You will need the Netlify Command Line Interface (CLI) to deploy your code. If you haven't already, install it by running:

bash Copy code

npm install -g netlify-cli Configure Netlify:

# Run the following command to log in to your Netlify account and link your repository:

bash Copy code

netlify login

Follow the prompts to log in and authorize the Netlify CLI to access your account.

# Create a netlify.toml File:

Create a netlify.toml file in the root of your project. This file is used to configure your build settings. Here's an example netlify.toml for a Node.js- based serverless function:

toml Copy code [build]

functions = "functions"

This assumes your serverless functions are located in a directory named "functions."

# Deploy Your API:

Deploy your API to Netlify using the following command: bash

Copy code netlify deploy

# Create MongoDB Database for the Laptop Shopping Cart.

**Solution:**

# User Collection Schema

{

name: {

type: String, required: true,

},

email: {

type: String, required: true, unique: true,

},

date: {

type: Date, default: Date.now,

},

}

# Laptop Collection Schema

title:{

type:String, required:true

},

price:{

type:Number, required:true

},

count:{type:Number}, url:{type:String}, description:{type:String}

}

db.createCollection("laptops") db.createCollection("users")

# Connect the Laptop Shopping Cart API with MongoDB Database.

**Backend Code:**

//Package.json

{

"name": "backend",

"version": "1.0.0",

"description": "",

"main": "index.js", "scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"keywords": [],

"author": "",

"license": "ISC", "dependencies": {

"cors": "^2.8.5",

"express": "^4.18.2",

"mongodb": "^6.1.0",

"mongoose": "^7.6.2"

}

}

//index.js

// To connect with your mongoDB database const mongoose = require('mongoose');

mongoose.connect('mongodb://127.0.0.1:27017/', { dbName: 'LaptopTest',

useNewUrlParser: true, useUnifiedTopology: true

}).then(()=>console.log('Connected to database'))

.catch((err)=>console.log(err));

// Schema for users of app

const UserSchema = new mongoose.Schema({ name: {

type: String, required: true,

},

email: {

type: String, required: true, unique: true,

},

date: {

type: Date

default: Date.now,

},

});

const User = mongoose.model('User', UserSchema); User.createIndexes();

// For backend and express

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

const cors = require("cors"); console.log("App listen at port 5000"); app.use(express.json()); app.use(cors());

app.get("/", (req, resp) => { resp.send("App is Working");

});

app.post("/register", async (req, resp) => { try {

const user = new User(req.body); let result = await user.save(); result = result.toObject();

if (result) {

delete result.password; resp.send(req.body); console.log(result);

} else {

console.log("User already register");

}

} catch (e) {

resp.send("Something Went Wrong");

}

});

app.listen(5000);

# FrontEnd Code:

//App.js in React

import { useState } from 'react' export default function TestLogin() {

const [name, setName] = useState(""); const [email, setEmail] = useState(""); const handleOnSubmit = async (e) => {

e.preventDefault();

let result = await fetch( 'http://127.0.0.1:5000/register', {

method: "post",

body: JSON.stringify({ name, email }), headers: {

'Content-Type': 'application/json'

}

})

result = await result.json(); console.warn(result);

if (result) {

alert("Data saved succesfully"); setEmail("");

setName("");

}

}

return (

<>

</>

);

}

<h1>This is React WebApp </h1>

<form action="">

<input type="text" placeholder="name"

value={name} onChange={(e) => setName(e.target.value)} />

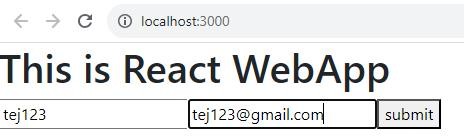
<input type="email" placeholder="email"

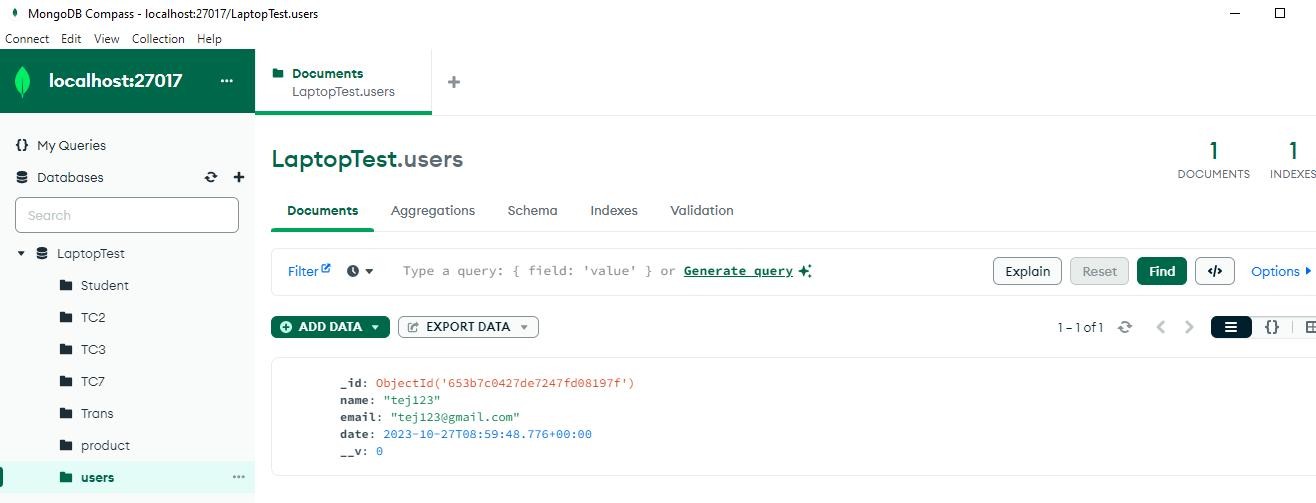
value={email} onChange={(e) => setEmail(e.target.value)} />

<button type="submit" onClick={handleOnSubmit}>submit</button>

</form>

# Output:





1. **Create the following Web Pages and make Page Navigation using Angular for the Laptop Shopping Cart.**

# Home About Contact MyCart Login/Signup

**//app-routing.module.ts**

import { NgModule } from '@angular/core';

import { RouterModule, Routes } from '@angular/router'; import { HomeComponent } from './home/home.component'; import { AboutComponent } from './about/about.component';

import { ContactComponent } from './contact/contact.component'; const routes: Routes = [

{path:'',component:HomeComponent},

{path:'about',component:AboutComponent},

{path:'contact',component:ContactComponent}

];

@NgModule({

imports: [RouterModule.forRoot(routes)], exports: [RouterModule]

})

export class AppRoutingModule { }

# //app.component.html

<app-header></app-header>

<div class="container">

<router-outlet></router-outlet>

</div>

<app-footer></app-footer>

# //home.component.html

<div class="container">

<div class="row">

<div class="col-sm">

<div class="card">

<img src="/assets/Dell\_1.jpg" width="300px" height="300px">

<label><b>Dell\_1</b></label> Rs.50690/-

<button type="button" class="btn btn-warning">Add to MyCart</button>

</div>

</div>

<div class="col-sm">

<div class="card">

<img src="/assets/Acer\_1.jpg" width="300px" height="300px">

<label><b>Acer\_1</b></label> Rs.45500/-

<button type="button" class="btn btn-warning">Add to MyCart</button>

</div>

</div>

<div class="col-sm">

<div class="card">

<img src="/assets/Hp\_1.jpg" width="300px" height="300px">

<label><b>Hp\_1</b></label> Rs.55000/-

<button type="button" class="btn btn-warning">Add to MyCart</button>

</div>

</div>

</div>

</div>

<br/>

# //header.component.html

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<div class="container-fluid">

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" routerLink="" >Home</a>

</li>

<li class="nav-item">

<a class="nav-link" routerLink="./about">Aboutus</a>

</li>

<li class="nav-item">

<a class="nav-link" routerLink="./contact">Contact</a>

</li>

<li class="nav-item">

<a class="nav-link" href="">MyCart</a>

</li>

<li class="nav-item">

<a class="nav-link" href="">Login</a>

</li>

</ul>

</div>

</div>

</nav>

# //about.component.html

<p>about Page!</p>

<br>

<p>about Page!</p>

# //contact.component.html

<p>contact Page!</p>

<br>

<p>contact Page!</p>

# //footer.component.html

<p>Created By Computer Engineering Department!</p>

# Output:

