Login Application

Backend folder structure

-.env

-db.py

-auth.py

-model.py

-main.py

Packages

Pip install fastapi

Pip install redis

Pip install dotenv

Pip install Basemodel

pip install pydantic[email]

pip install "passlib[bcrypt]"

Coding for each page

**.env**

MONGO\_URI=mongodb://localhost:27017

REDIS\_HOST=localhost

REDIS\_PORT=6379

**Db.py**

from pymongo import MongoClient

import redis

import os

from dotenv import load\_dotenv

load\_dotenv()

# MongoDB setup

mongo\_client = MongoClient(os.getenv("MONGO\_URI"))

mongo\_db = mongo\_client["myapp"]

user\_collection = mongo\_db["users"]

# Redis setup

redis\_client = redis.Redis(

    host=os.getenv("REDIS\_HOST"),

    port=int(os.getenv("REDIS\_PORT")),

    decode\_responses=True

)

Model.py

from pydantic import BaseModel, EmailStr

class UserRegister(BaseModel):

    name: str

    email: EmailStr

    password: str

class UserLogin(BaseModel):

    email: EmailStr

    password: str

class UserOut(BaseModel):

    name: str

    email: EmailStr

auth.py

from passlib.context import CryptContext

import uuid

pwd\_context = CryptContext(schemes=["bcrypt"], deprecated="auto")

def hash\_password(password: str):

    return pwd\_context.hash(password)

def verify\_password(password: str, hashed: str):

    return pwd\_context.verify(password, hashed)

def create\_session\_id():

    return str(uuid.uuid4())

main.py

from fastapi import FastAPI, HTTPException, Depends, Header

from db import user\_collection, redis\_client

from models import UserRegister, UserLogin, UserOut

from auth import hash\_password, verify\_password, create\_session\_id

from fastapi.responses import JSONResponse

from fastapi.middleware.cors import CORSMiddleware

import json

app = FastAPI()

app.add\_middleware(

    CORSMiddleware,

    allow\_origins=["\*"],  # Or restrict to ["http://localhost"]

    allow\_credentials=True,

    allow\_methods=["\*"],

    allow\_headers=["\*"],

)

@app.post("/register")

def register(user: UserRegister):

    if user\_collection.find\_one({"email": user.email}):

        raise HTTPException(status\_code=400, detail="Email already exists")

    user\_data = user.dict()

    user\_data["password"] = hash\_password(user\_data["password"])

    user\_collection.insert\_one(user\_data)

    return {"message": "Registration successful"}

@app.post("/login")

def login(data: UserLogin):

    user = user\_collection.find\_one({"email": data.email})

    if not user or not verify\_password(data.password, user["password"]):

        raise HTTPException(status\_code=401, detail="Invalid credentials")

    session\_id = create\_session\_id()

    redis\_client.setex(f"session:{session\_id}", 3600, user["email"])

    return {"message": "Login successful", "session\_id": session\_id}

def get\_current\_email(session\_id: str = Header(...)):

    email = redis\_client.get(f"session:{session\_id}")

    if not email:

        raise HTTPException(status\_code=403, detail="Invalid or expired session")

    return email

@app.get("/profile", response\_model=UserOut)

def profile(email: str = Depends(get\_current\_email)):

    cached = redis\_client.get(f"profile:{email}")

    if cached:

        return json.loads(cached)

    user = user\_collection.find\_one({"email": email}, {"\_id": 0, "name": 1, "email": 1})

    if not user:

        raise HTTPException(status\_code=404, detail="User not found")

    redis\_client.setex(f"profile:{email}", 60, json.dumps(user))

    return user

Front-end

Folder structure

-index.html

-register.html

-profile.html

Register.html

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <title>Register</title>

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" />

  <style>

    body {

      background: linear-gradient(to right, #ff512f, #dd2476);

      height: 100vh;

      display: flex;

      justify-content: center;

      align-items: center;

    }

    .register-box {

      background: #fff;

      padding: 2rem;

      border-radius: 1rem;

      box-shadow: 0 0 15px rgba(0, 0, 0, 0.2);

      width: 100%;

      max-width: 500px;

    }

  </style>

</head>

<body>

  <div class="register-box">

    <h2 class="text-center mb-4">Register</h2>

    <form id="registerForm">

      <div class="mb-3">

        <label for="name" class="form-label">Full Name</label>

        <input type="text" class="form-control" id="name" required />

      </div>

      <div class="mb-3">

        <label for="email" class="form-label">Email address</label>

        <input type="email" class="form-control" id="email" required />

      </div>

      <div class="mb-3">

        <label for="password" class="form-label">Password</label>

        <input type="password" class="form-control" id="password" required />

      </div>

      <div class="mb-3">

        <label for="confirmPassword" class="form-label">Confirm Password</label>

        <input type="password" class="form-control" id="confirmPassword" required />

      </div>

      <button type="submit" class="btn btn-danger w-100">Register</button>

      <div id="message" class="mt-3 text-center"></div>

    </form>

      Already Registered <a style="text-align: center;" href="index.html">SignIn</a>

  </div>

  <script>

    document.getElementById("registerForm").addEventListener("submit", async function (e) {

      e.preventDefault();

      const name = document.getElementById("name").value;

      const email = document.getElementById("email").value;

      const password = document.getElementById("password").value;

      const confirmPassword = document.getElementById("confirmPassword").value;

      const messageBox = document.getElementById("message");

      if (password !== confirmPassword) {

        messageBox.innerText = "Passwords do not match!";

        messageBox.className = "text-danger text-center";

        return;

      }

      try {

        const response = await fetch("http://localhost:8000/register", {

          method: "POST",

          headers: {

            "Content-Type": "application/json"

          },

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

        });

        const result = await response.json();

        if (response.ok) {

          messageBox.className = "text-success text-center";

          messageBox.innerText = result.message || "Registered successfully!";

          // Optional: Redirect to login

           setTimeout(() => window.location.href = "/index.html", 2000);

        } else {

          messageBox.className = "text-danger text-center";

          messageBox.innerText = result.message || "Registration failed.";

        }

      } catch (err) {

        messageBox.className = "text-danger text-center";

        messageBox.innerText = "Server not reachable.";

      }

    });

  </script>

</body>

</html>

Index.html

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <title>Login</title>

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" />

  <style>

    body {

      background: linear-gradient(to right, #6a11cb, #2575fc);

      height: 100vh;

      display: flex;

      justify-content: center;

      align-items: center;

    }

    .login-box {

      background: #fff;

      padding: 2rem;

      border-radius: 1rem;

      box-shadow: 0 0 15px rgba(0, 0, 0, 0.2);

      width: 100%;

      max-width: 400px;

    }

  </style>

</head>

<body>

  <div class="login-box">

    <h2 class="text-center mb-4">Login</h2>

    <form id="loginForm">

      <div class="mb-3">

        <label for="email" class="form-label">Email address</label>

        <input type="email" class="form-control" id="email" required />

      </div>

      <div class="mb-3">

        <label for="password" class="form-label">Password</label>

        <input type="password" class="form-control" id="password" required />

      </div>

      <button type="submit" class="btn btn-primary w-100">Login</button><br/>

      <div id="message" class="mt-3 text-center text-danger"></div>

    </form>

          New User <a style="text-align: center;" href="register.html">Signup</a>

  </div>

  <p id="message"></p>

  <script>

    document.getElementById("loginForm").addEventListener("submit", async function (e) {

      e.preventDefault(); // Prevent default form submission

      const email = document.getElementById("email").value;

      const password = document.getElementById("password").value;

      try {

        const response = await fetch("http://localhost:8000/login", {

          method: "POST",

          headers: {

            "Content-Type": "application/json"

          },

          body: JSON.stringify({ email, password })

        });

        if (response.ok) {

           const data = await response.json();

    localStorage.setItem("session\_id", data.session\_id);  // Save session

    window.location.href = "profile.html";

  } else {

    alert("Login failed");

  }

      } catch (err) {

        document.getElementById("message").innerText = "Server not reachable.";

      }

    });

  </script>

</body>

</html>

Profile.html

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>User Profile</title>

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

  <!-- Bootstrap 5 CDN -->

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body class="bg-light">

  <div class="container mt-5">

    <div class="card shadow p-4">

      <h2 class="mb-4">Profile</h2>

      <div id="profileDetails">

        <p><strong>Name:</strong> <span id="name">Loading...</span></p>

        <p><strong>Email:</strong> <span id="email">Loading...</span></p>

      </div>

      <div id="error" class="text-danger mt-3" style="display: none;"></div>

    </div>

  </div>

  <script>

    const sessionId = localStorage.getItem("session\_id"); // Assume stored after login

    console.log(sessionId);

    fetch("http://127.0.0.1:8000/profile", {

      method: "GET",

      headers: {

        "session-id": sessionId

      }

    })

    .then(response => {

        console.log(response)

      if (!response.ok) throw new Error("Session expired or unauthorized.");

      return response.json();

    })

    .then(data => {

      document.getElementById("name").textContent = data.name;

      document.getElementById("email").textContent = data.email;

    })

    .catch(err => {

      document.getElementById("error").textContent = err.message;

      document.getElementById("error").style.display = "block";

      document.getElementById("profileDetails").style.display = "none";

    });

  </script>

</body>

</html>