



MAGNET LABS



T3A3 PRESENTATION

Mostofa, Trenton & Perri



Welcome to Magnet Labs

Magnet Labs is a digital marketing and web development agency. This app will function as both a company website and a client portal. Clients will be able to manage their projects, receive updates, and submit feedback, while admins will have tools to oversee client accounts, publish announcements, and handle communications.

THE TEAM!

SHEKH MOSTOFA ABEDIN:

Product Owner, responsible for overall project vision, setup of environments, infrastructure (DevOps tasks), database administration, and backend development.

PERRI ADKINS:

Resident UX/UI Designer and Frontend Development all-rounder, handling all aspects of the user interface, user experience, and visual design.

TRENTON DUNSTAN:

Backend Development all-rounder, focusing on application architecture, data flow design, API integration, and backend logic.

THE APP !



MAGNET LABS™

Our Services Our Work Contact Us

THE ART OF DIGITAL ATTRACTION

We craft solutions that **Attract**, **Engage**, & **Convert**.

OUR SERVICES

We provide cutting-edge digital solutions to help your business grow.

WEB DEVELOPMENTSEO OPTIMISATIONSOCIAL MEDIA MARKETING

OUR WORK

Take a look at some of our most exciting projects and case studies.

PROJECT 1PROJECT 2PROJECT 3

PROJECT 4PROJECT 5PROJECT 6

GET IN TOUCH

Have a question? Ready to start a project?

Let's connect!

Contact Us

© 2025 Magnet Labs. All rights reserved. [Facebook](#) [Instagram](#) [LinkedIn](#)

Services Page

MAGNET LABS™

Our Services Our Work Contact Us

OUR EXPERTISE, YOUR ADVANTAGE

From cutting-edge web development to impactful marketing strategies, we deliver solutions that help your business stand out.

CUSTOM WEB DEVELOPMENT

We create responsive, visually stunning, and functional websites tailored to your brand.

- Mobile-friendly design
- SEO-optimized structure
- Custom integrations (e.g., e-commerce, CRM tools)



SEO THAT DRIVES RESULTS

Boost your online visibility with tailored strategies that get your business noticed.

- Keyword research and optimization
- On-page and off-page SEO
- Monthly performance reporting



SOCIAL MEDIA MARKETING

Create a strong social presence with campaigns that captivate your audience.

- Platform-specific strategies (e.g., Instagram, LinkedIn, Facebook)
- Professional content creation and curation
- Analytics and performance tracking to measure success



TAILORED PACKAGES FOR **EVERY** BUSINESS

Choose from flexible packages combining our core services to suit your needs.

STARTER PACKAGE

Web Development
Basic SEO

GROWTH PACKAGE

Web Development
SEO
Social Media Marketing

CUSTOM PACKAGE

Fully tailored to your business goals

CONTACT US FOR PRICING

GET IN TOUCH

Have a question? Ready to start a project?

Let's connect!

Contact Us

© 2025 Magnet Labs. All rights reserved. 



OUR WORK SPEAKS FOR ITSELF

Explore the projects that showcase our expertise and creativity.

PROJECT TITLE 1



A brief description of this project.

[LEARN MORE](#)

PROJECT TITLE 2



Another project showcasing our expertise.

[LEARN MORE](#)

PROJECT TITLE 3



Discover the innovative solutions we deliver.

[LEARN MORE](#)

PROJECT TITLE 4



Explore how we turn ideas into success.

[LEARN MORE](#)

PROJECT TITLE 5



A remarkable project crafted with precision.

[LEARN MORE](#)

PROJECT TITLE 6



Elevating brands through creativity and strategy.

[LEARN MORE](#)

GET IN TOUCH

Have a question? Ready to start a project?

Let's connect!

[Contact Us](#)



LET'S BUILD SOMETHING GREAT TOGETHER

Contact us for amazing web solutions!

0412 345 678

hello@magnetlabs.com.au

MESSAGE US

Your Name

Your Email

Starter Package

Your Message

Send Message



FREQUENTLY ASKED QUESTIONS

What services does Magnet Labs provide?

+

How can I get started with Magnet Labs?

-

Simply use the contact form on this page or email us at info@magnetlabs.com. Our team will get back to you within 1-2 business days.

Can I customize my package?

+

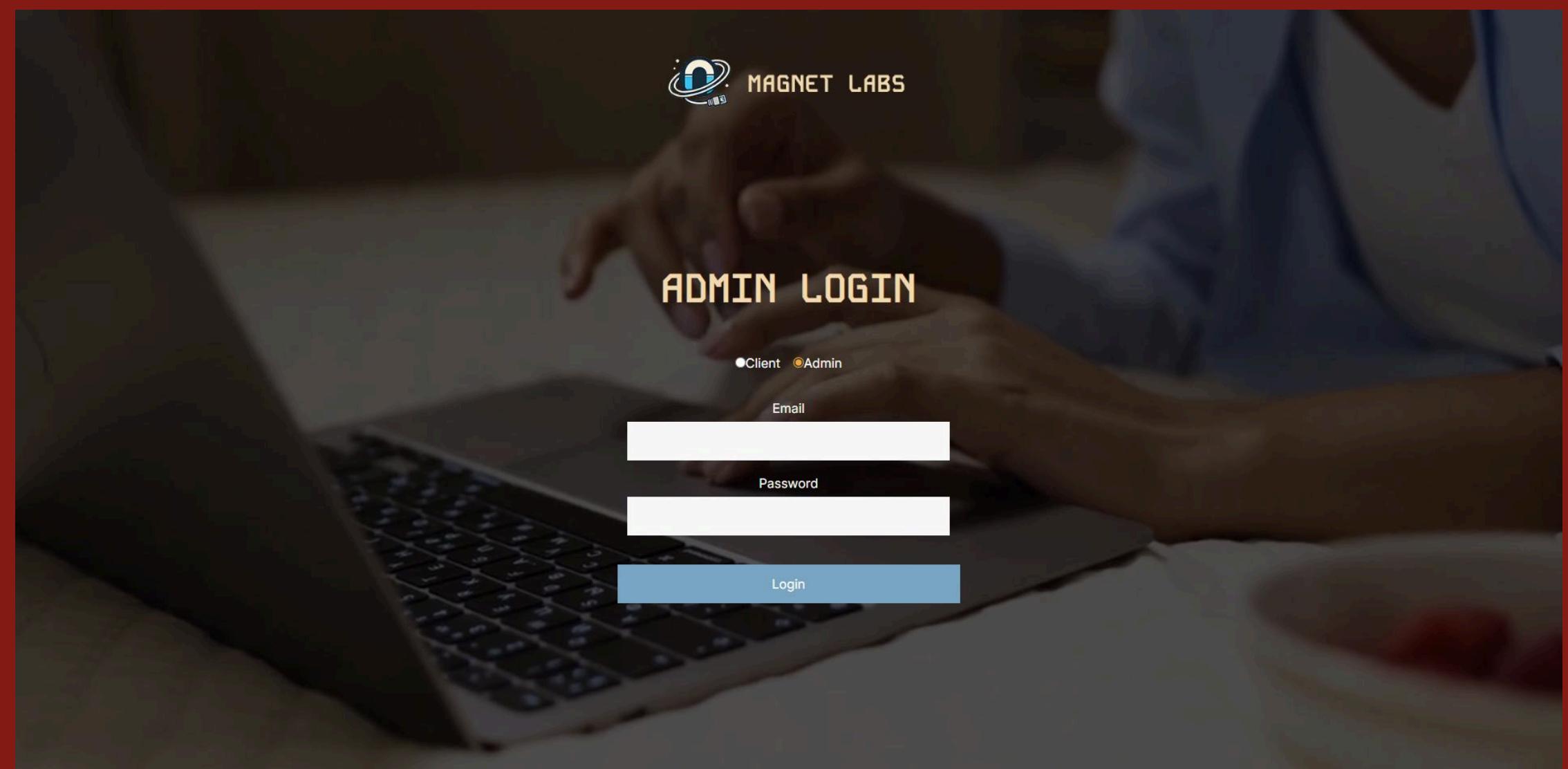
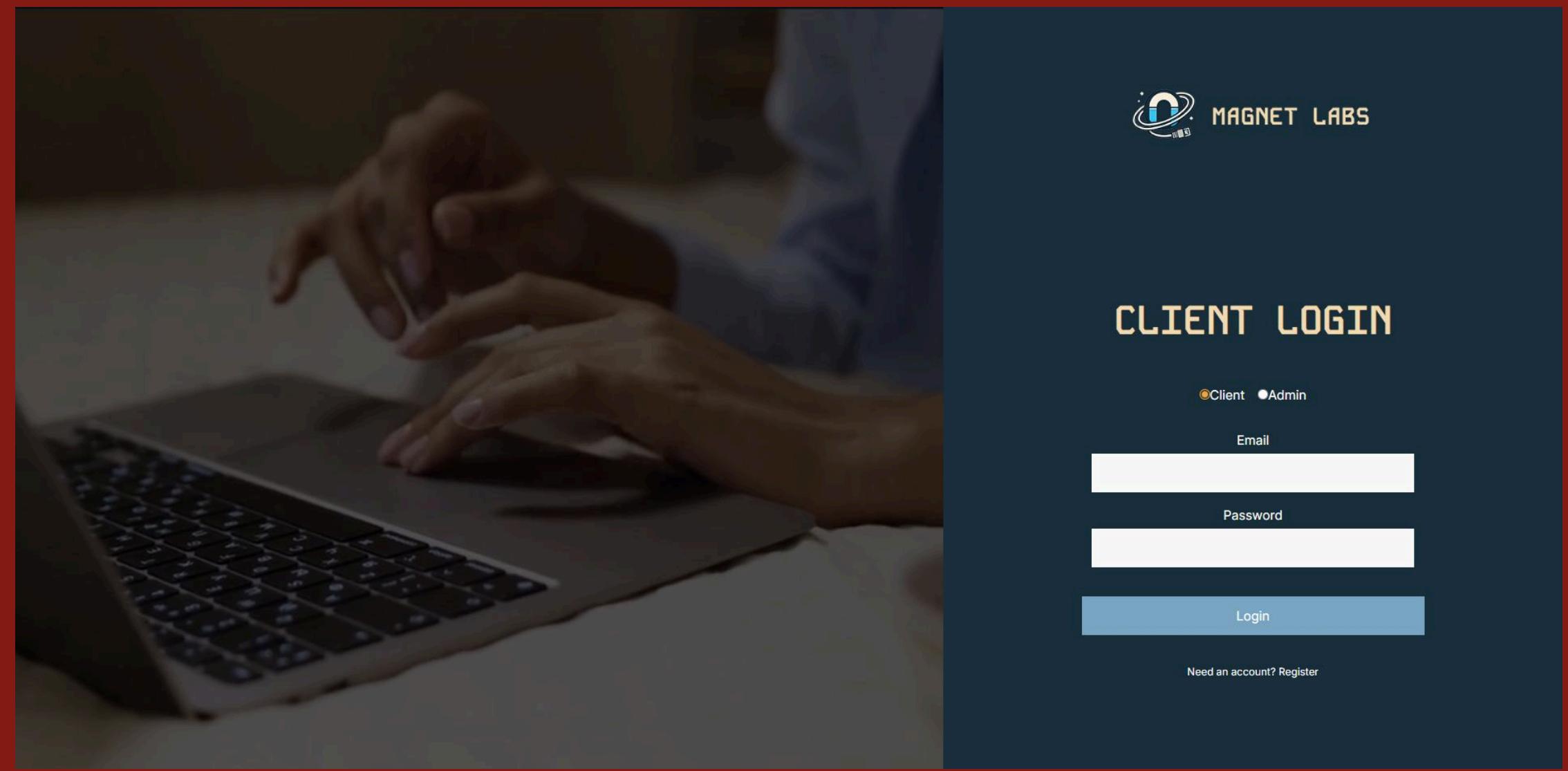
What industries do you specialize in?

+

Do you offer ongoing support?

+

Login page (Admin & Client Side-by-side)



Registration Page



NEW CLIENT ACCOUNT

Client Admin

Name

Email

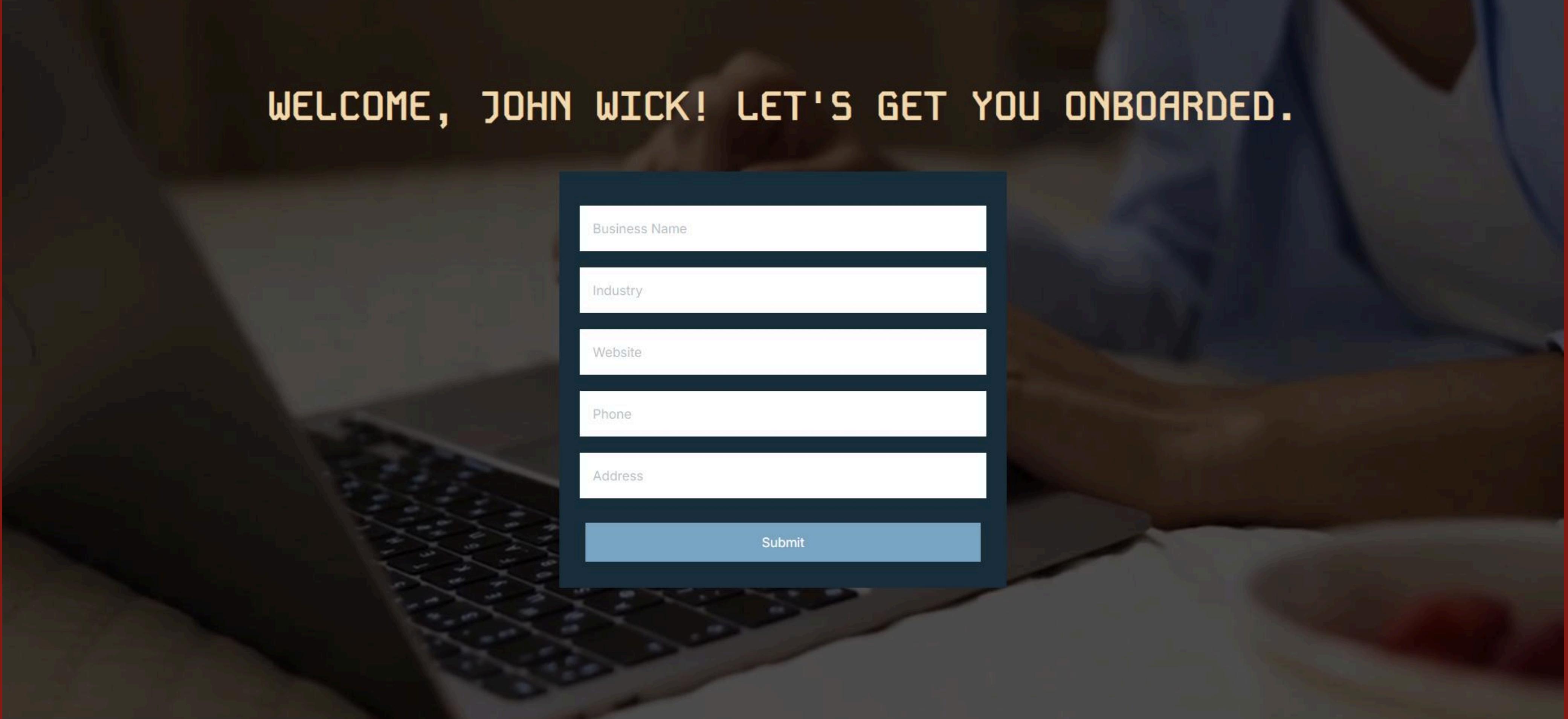
Password

Register

Already have an account? [Login](#)

Onboarding Form

WELCOME, JOHN WICK! LET'S GET YOU ONBOARDED.



A dark blue rectangular form overlay on a blurred background of a laptop keyboard. The form contains five input fields with labels: 'Business Name', 'Industry', 'Website', 'Phone', and 'Address'. Below these fields is a light blue 'Submit' button.

Business Name
Industry
Website
Phone
Address
Submit

CLIENT DASHBOARD

My Business Details

Business Name: Tarasov Mob
Industry: Extortion
Website: https://www.baba_yaga.com
Phone: 3105648005
Address: 123 Kill Street

[Edit Business](#)

My Projects

- Project Name:** Talk to Viggo
Status: In Progress
Description: Just talk to him
Start Date: 6/1/2024
End Date: 12/31/2024
- Project Name:** Get Losef
Status: In Progress
Description: Get em
Start Date: 6/1/2024
End Date: 12/31/2024

Announcements

- Title:** New Announcement
Content: We are excited to announce our new feature launch!
Active: Yes
Date: 3/13/2025
- Title:** Test2
Content: sdasda
Active: Yes
Date: 3/13/2025
- Title:** test announcement
Content: test announcement for business
Active: Yes
Date: 3/13/2025

CSS Fixes in progress!

Admin Portal

ADMIN DASHBOARD

Welcome, Admin!

MANAGE PROJECTS

Back to Dashboard

ALL PROJECTS

Add Project

Test Project Client: testUser@test.com Due Date: 20/03/2025 Edit Delete	Talk to Viggo Client: John Wick Due Date: 31/12/2024 Edit Delete	Get Losef Client: John Wick Due Date: 31/12/2024 Edit Delete
-------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------

MANAGE CLIENTS

Back to Dashboard

All Clients

Add Client

testClientUser@test.com Business Name: N/A Phone: N/A Email: testClientUser@test.com Edit Delete	testUser@test.com Business Name: N/A Phone: N/A Email: testUser@test.com Edit Delete	Tester Business Name: N/A Phone: N/A Email: Tester@gmail.com Edit Delete
John Wick Business Name: N/A Phone: N/A Email: Baba_yaga@gmail.com Edit Delete		

MANAGE ANNOUNCEMENTS

Back to Dashboard

ALL ANNOUNCEMENTS

Add Announcement

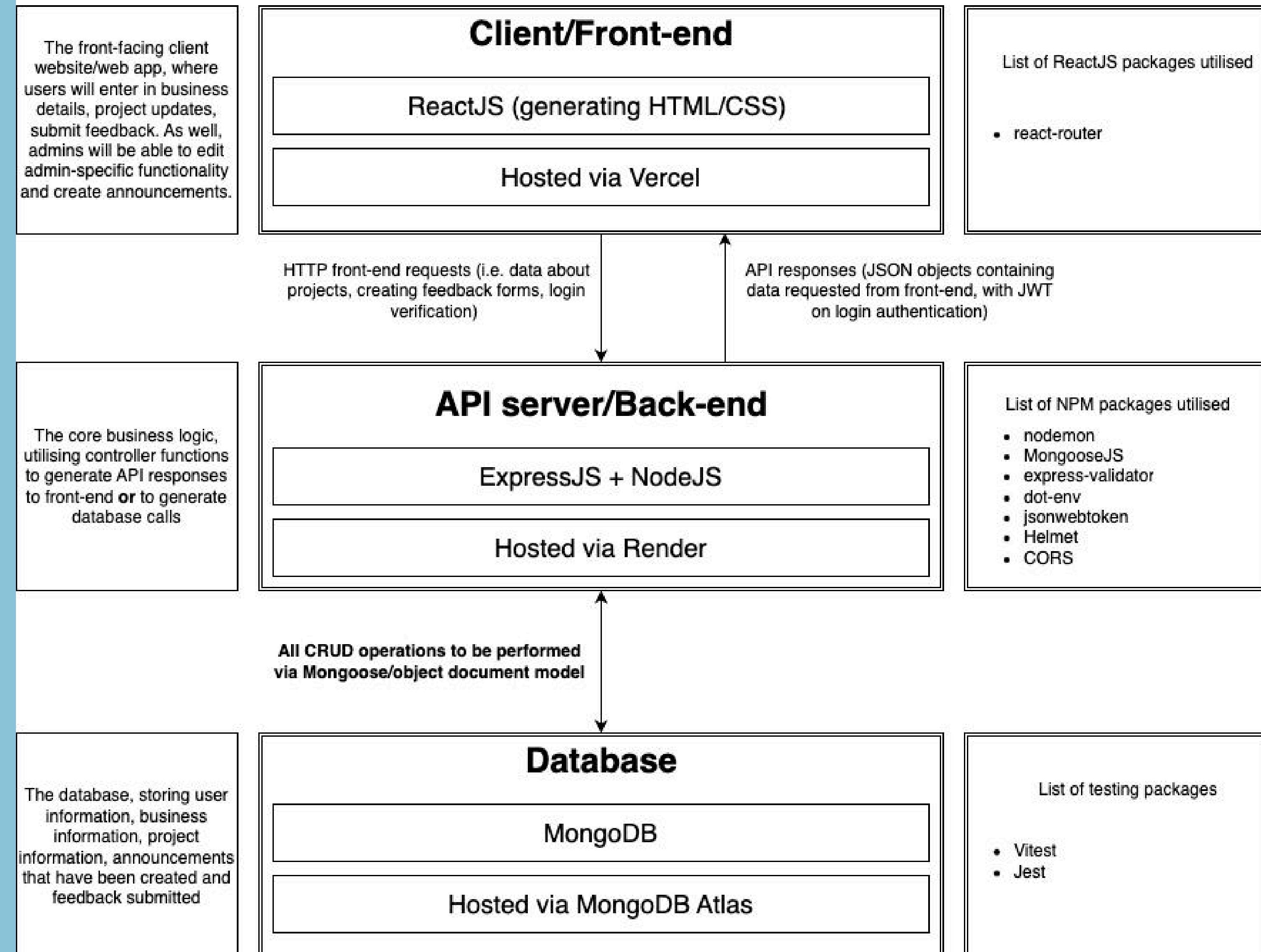
New Announcement Example Change We are excited to announce our new feature launch! Date: 13/03/2025 Edit Delete	Test2 Example Change sdasda Date: 13/03/2025 Edit Delete	test announcement testBusiness test announcement for business Date: 13/03/2025 Edit Delete
--------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------

CSS Fixes in progress!

APP ARCHITECTURE OVERVIEW

Magnet Labs Architecture Diagram

Magnet Labs is a digital marketing and web development agency. This app will function as both a company website and a client portal, utilising a modern front-end to display details and information about the user's company with information being retrieved from back-end server connected to a database to perform CRUD (create, read, update and delete) operations.



ESSENTIAL CODE

Frontend - Login API Request (Network Request)

We start with

File: frontend/src/components/loginForm/loginForm.jsx

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

    const data = await response.json();
    if (response.ok) {
      localStorage.setItem("authToken", data.token); // Store JWT for authentication
      localStorage.setItem("userId", data.user._id); // Store user ID for future use
      navigate("/dashboard"); // Redirect user after login
    } else {
      console.error("Login failed:", data.message);
    }
  } catch (error) {
    console.error("Error during login:", error);
  }
};
```

Key Takeaways:

- Network request (fetch("/login")) → Sends user credentials to backend.
- JWT is stored (localStorage.setItem("authToken", data.token)) → Used for authentication in future API calls.
- Conditional logic (if (response.ok)) → Redirects user on success, logs error otherwise

Backend - Login Route (Auth Route)

File: **backend/src/routes/authRoutes.js**

```
import express from "express";
import { loginUser } from "../controllers/authController.js"; // Import controllers

const router = express.Router();

// Route: Login a new user
router.post("/", loginUser);

export default router;
```

Key Takeaways:

Defines a POST /login route.

Calls loginUser from authController.js (separation of concerns).

Middleware could be added for rate limiting or logging.

Backend - Verifying User & Generating JWT (Authentication Logic)

File: backend/src/controllers/authController.js

```
import User from "../models/User.js";
import bcrypt from "bcrypt";
import jwt from "jsonwebtoken";

const loginUser = async (req, res) => {
  const { email, password } = req.body;
  const user = await User.findOne({ email });

  if (!user) {
    return res.status(400).json({ message: "Invalid Credentials." });
  }

  const isPasswordValid = await bcrypt.compare(password, user.password);
  if (!isPasswordValid) {
    return res.status(400).json({ message: "Wrong password." });
  }

  const token = jwt.sign(
    { userID: user._id, role: user.role }, // Include user role in JWT token
    process.env.JWT_SECRET || "secret",
    { expiresIn: "1h" }
  );

  res.json({ token, user, role: user.role }); // Send token & user data to frontend
};

export { loginUser };
```

**Database operation (User.findOne({ email })) →
Fetches user from MongoDB.**

Conditional logic (if (!user) return res.status(400)) → Handles login errors.

Password validation (bcrypt.compare(password, user.password)) → Ensures correct credentials.

JWT creation (jwt.sign()) → Secure token issued for authentication.

Middleware - JWT Validation & Role-Based Access Control

File: backend/src/middlewares/authMiddleware.js

```
import jwt from "jsonwebtoken";

const authenticateUser = (req, res, next) => {
  const token = req.headers.authorization ? req.headers.authorization.substring("Bearer ".length) : "";
  if (!token) {
    return res.status(401).json({ message: 'Access Denied. No token provided.' });
  }

  try {
    const decoded = jwt.verify(token, process.env.JWT_SECRET || "secret");
    req.user = decoded; // Attach decoded user info to `req`
    next();
  } catch (error) {
    res.status(403).json({ message: 'Invalid Token.' });
  }
};
```

```
const authorizeAdmin = (req, res, next) => {
  if (req.user.role !== 'admin') {
    return res.status(403).json({ message: 'Access Denied. Admins only.' });
  }
  next();
};
```

Key Takeaways:

- Extracts JWT from request headers (req.headers.authorization).
- Verifies token using jwt.verify() → Ensures valid authentication.
- Attaches decoded user data (req.user = decoded) for API access.
- Handles unauthorized access (return res.status(401)).

Key Takeaways:

- Checks user role (if (req.user.role !== 'admin')) before granting access.
- Restricts sensitive actions (e.g., creating/deleting users, managing data).
- Allows only admins to perform admin-level operations.

Why This Matters - Authentication & Security

- Prevents unauthorised access to protected API routes.
- Ensures only admins can perform administrative actions.
- JWT allows stateless authentication → No need to store session data on the server.
- Enhances security by encrypting passwords & validating tokens before granting access.

OUR CHALLENGES

COREUI.IO INTEGRATION

```
.header { // Basic classnames (these were already used by global styling in CoreUI.io integration)
  background-color: #192F3C !important;
  color: white;
  font-family: "Kode Mono", monospace;
  text-transform: uppercase;
  border: none;
  padding: 2rem 2.5rem;
  --cui-header-color: white; // Specific elements found using DevTools
  --cui-body-font-family: "Inter", sans-serif;
}
```

```
.header-text {
  color: #F3DCB2;
  font-size: 2rem;
}
```

```
.header-nav {
  margin: 0;
  width: 200px;
}
```



```
.dash-main-card {
  border-radius: 0% !important;
  margin-bottom: 2rem;
}
```

```
.dash-client-card {
  border-radius: 0% !important;
  margin-bottom: 2rem;
  cursor: pointer;
  padding: 1rem;
}
```

```
.dash-card-header { // Created unique identifiers for all elements
  to override global styles
  display: flex;
  justify-content: space-between;
  align-items: center;
  background-color: #D9D9D952;
  border: none;
  border-radius: 0% !important;
  font-family: "Kode Mono", monospace;
  font-weight: bold;
  text-transform: uppercase;
}
```

```
}
```

COREUI.IO INTEGRATION

Resources used:

CoreUi.io documentation:

<https://coreui.io/bootstrap/docs/2.1/getting-started/introduction/>

Chrome DevTools

MIDDLEWARE CHALLENGE

```
router.delete("/user/:id");
```

MIDDLEWARE CHALLENGE

```
// Middleware to authorize either the user themselves or an admin
const authUserOrAdmin = (req, res, next) => {
  // Extract token from Authorization header
  const token = req.headers.authorization ? req.headers.authorization.substring("Bearer ".length) : "";

  // If no token is found, deny access
  if (!token) {
    return res.status(401).json({ message: 'Access Denied. No token provided.' });
  }

  try {
    // Verify the token
    const decoded = jwt.verify(token, process.env.JWT_SECRET || "secret");
    req.user = decoded;

    // Allow the following:
    // Admins can edit any user or business profile
    // Users can edit their own profile (userID from token must match ID in the request parameter)
    // Users can update their own business details (userID from token must match the business owner's ID)
    if (
      req.user.role === 'admin' ||
      req.user.userID === req.params.id || // User updating their own profile
      req.user.userID === req.params.businessId // User updating their own business
    ) {
      next();
    } else {
      // Deny access if the user doesn't have permission
      return res.status(403).json({ message: 'Access Denied. You can only update your own profile or business.' });
    }
  } catch (error) {
    // Handle invalid tokens
    res.status(403).json({ message: 'Invalid Token.' });
  }
};

export { authUserOrAdmin };
```

MIDDLEWARE CHALLENGE

Resources used:

Express documentation:

<https://expressjs.com/en/guide/using-middleware.html>

GeekForGeeks:

<https://www.geeksforgeeks.org/middleware-in-express-js/>



MAGNET LABS

QUESTIONS?