

Lab Work 1: User Management System (Registration and Login)

Objective:

To create a basic user authentication system including user registration and login functionality.

Steps:

1. Create Database

For this lab, we used PostgreSQL as the relational database, with NextJS.

- Prisma was used as the ORM to define and interact with the database
- User model in schema.prisma:

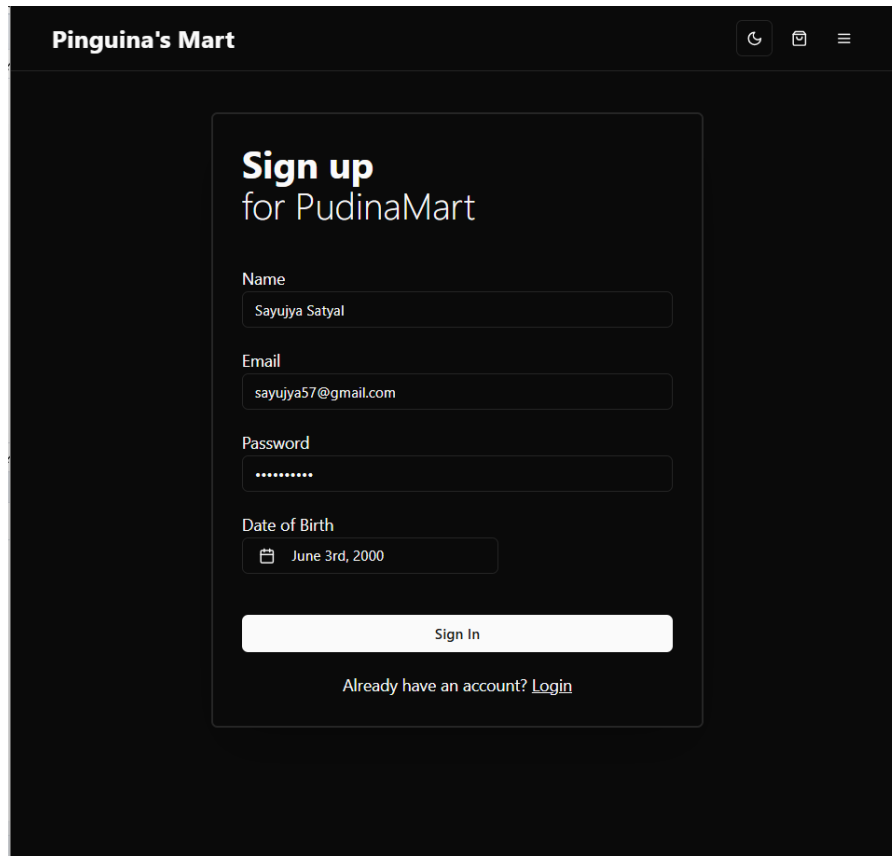
Field Name	Data Type	Details	What It Means
id	Text (ID)	Unique ID automatically created for each user	A special ID that identifies each user
name	Text	-	The user's full name
email	Text	Must be unique (no two users can have the same)	The user's email address
password	Text	-	The user's secret password (hidden/encrypted)
dob	Date	-	The user's date of birth
createdAt	Date & Time	Automatically set when the user is created	Records when the user signed up
cartItems	List of Carts	Connected to user's cart items	All the items the user added to their cart
reviews	List of Reviews	Connected to user's reviews	All the product reviews written by the user
orders	List of Orders	Connected to user's orders	All the orders placed by the user

2. Registration System

The registration logic is defined in a server action (auth.action.ts) and triggered from a form in the client side. The server action validates:

- Required fields
- Age restriction (must be ≥ 13)
- Password strength (using a utility function)
- Uniqueness of email
- If valid, the password is hashed using bcryptjs, and the user is saved in the PostgreSQL database via Prisma.

Outputs:



The image shows a web application interface for 'Pinguina's Mart'. The header includes the site name and navigation icons. The main content area is a registration form titled 'Sign up for PudinaMart'. It contains four input fields: 'Name' (filled with 'Sayuja Satyal'), 'Email' (filled with 'sayuja57@gmail.com'), 'Password' (filled with eight dots), and 'Date of Birth' (filled with a calendar icon and 'June 3rd, 2000'). Below the fields is a 'Sign In' button and a link 'Already have an account? Login'.

Pinguina's Mart

Sign up for PudinaMart

Name
Sayuja Satyal

Email
sayuja57@gmail.com

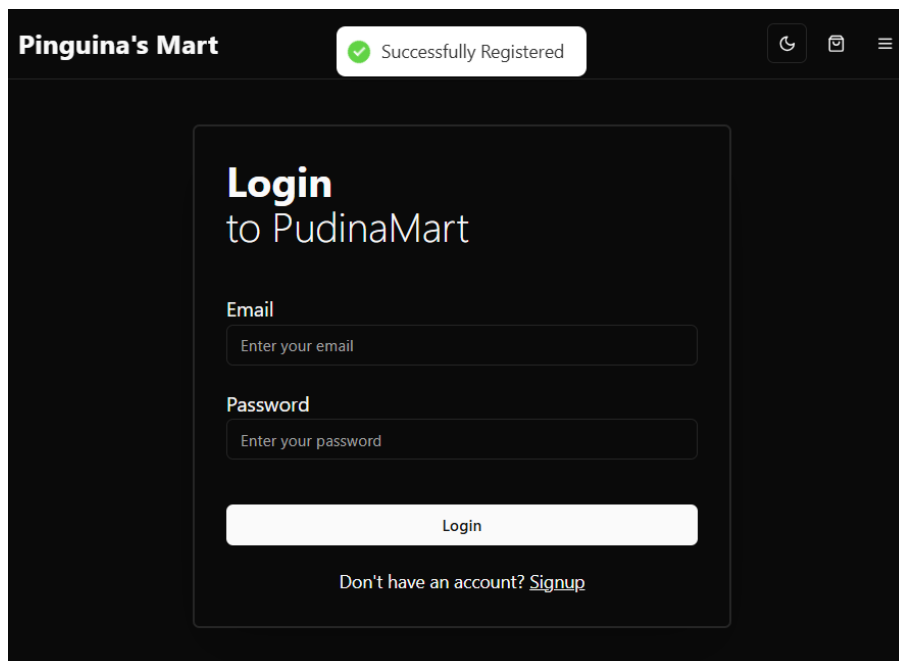
Password
.....

Date of Birth
June 3rd, 2000

Sign In

Already have an account? [Login](#)

Figure 1: Registration form for users



The image shows the same web application interface after successful registration. A green notification banner at the top says 'Successfully Registered'. The main content area is a login form titled 'Login to PudinaMart'. It contains two input fields: 'Email' (placeholder 'Enter your email') and 'Password' (placeholder 'Enter your password'). Below the fields is a 'Login' button and a link 'Don't have an account? Signup'.

Pinguina's Mart

Successfully Registered

Login to PudinaMart




Email
Enter your email

Password
Enter your password

Login

Don't have an account? [Signup](#)

Figure 2: Successful registration

Pinguina's Mart ✖ Email is already registered   

Sign up for PudinaMart

Name




Email

Password

Date of Birth

Already have an account? [Login](#)

Figure 3: Duplicate email registration

Pinguina's Mart ✖ You must be at least 13 years old to register.   

Sign up for PudinaMart

Name

Email

Password

Date of Birth

Already have an account? [Login](#)

Figure 4: Age validation during registration

	id [PK] text	name text	email text	password text	dob timestamp without time zone (3)	createdAt timestamp without time zone (3)
1	cmbpbtr20000t5oi88zdumg	Sayujya Satyal	sayujya57@gmail.com	\$2b\$10\$/9qJbdsxIEPMMTk5fdzL1e5PBwQLwYYB3XmDolYcxVul8RCbwcl...	2000-06-02 18:15:00	2025-06-09 16:48:35.533

Figure 5: Stored data in database after registration

3. Login System:

- The login form accepts email and password and sends a request to the /api/login route.
- On the server:
 - Credentials are validated against stored hashes.
 - A JWT token is issued on success and stored in an HTTP-only cookie.

Outputs:

Pinguina's Mart

Login
to PudinaMart

Email
sayujya57@gmail.com

Password
.....

Login

Don't have an account? [Signup](#)

Figure 6: Login form

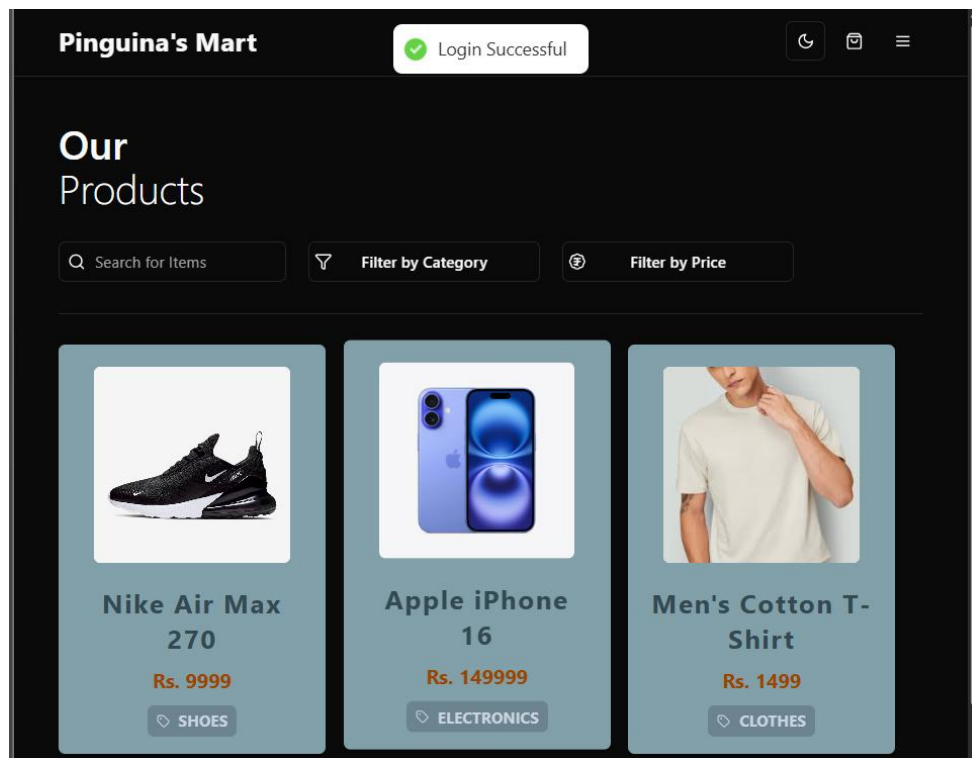


Figure 7: Successful login

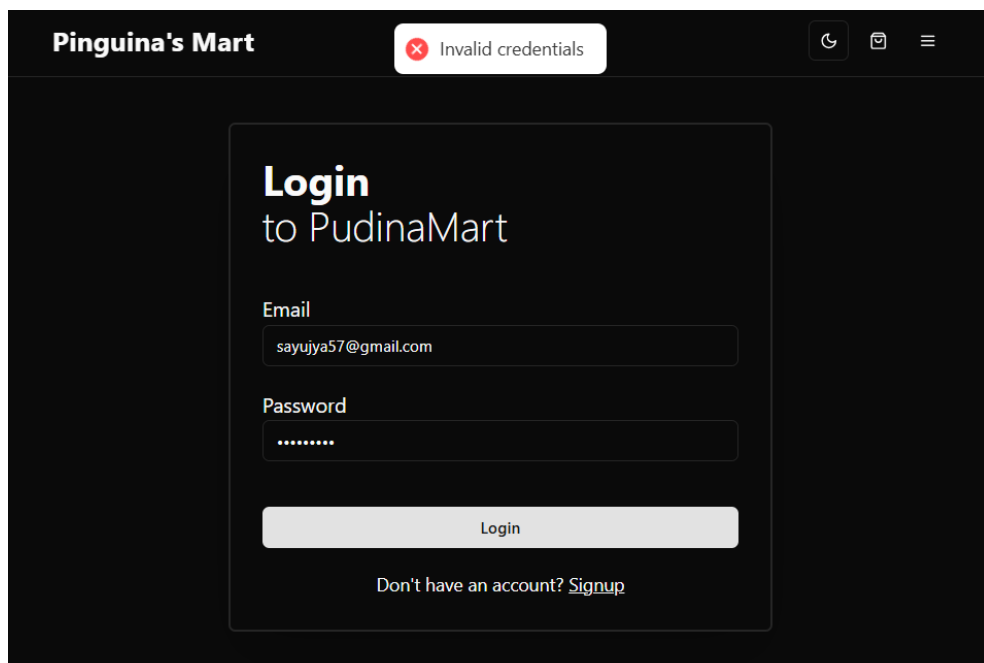


Figure 8: Invalid credentials using login

