

ଲେନ-ଦେନ

A Web-Based Online Classifieds Platform

Introduction:

Len-Den is a web-based online classifieds platform that allows users to buy and sell items securely within a single application. The platform is inspired by real-world marketplaces such as OLX and Quikr, providing a simple and intuitive interface for listing items, browsing available products, and managing user profiles. The project is built using the **MERN stack (without React)** and focuses on practical implementation of authentication, database management, and RESTful APIs.

Objectives:

The main objectives of this project are:

- To create a secure buy–sell marketplace
- To implement user authentication using sessions
- To allow users to list, cancel, and purchase items
- To maintain user statistics (items bought, sold, selling)
- To design a clean and responsive user interface
- To gain hands-on experience with full-stack development

Features:

User Authentication

- Provides user registration with secure password storage using bcrypt based hashing
- Implements user login through session-based authentication
- Supports secure logout functionality by destroying active user sessions
- Ensures restricted routes are accessible only to authenticated users

Buying and Selling

- Enables users to list items by providing an image, price, category, and description
- Handles image uploads using Multer middleware
- Displays all available items in a responsive grid-based layout
- Prevents users from purchasing items listed by themselves
- Automatically removes purchased items from active listings
- Updates buyer and seller statistics upon successful transactions

Profile Management

- Displays the authenticated user's username and profile image
- Presents user activity statistics, including:
 - Number of items bought
 - Number of items sold
 - Number of items currently listed for sale
- Allows users to cancel their listed items
- Provides a logout option for secure termination of user sessions

Filtering and Sorting

- Allows filtering of items based on predefined categories
- Supports sorting of items by price:
 - Low to High
 - High to Low
- Applies filtering and sorting dynamically without requiring page reloads

Security Features

- Ensures secure password handling through bcrypt hashing

- Implements session-based access control to maintain user authentication state
- Performs server-side validation to prevent unauthorized or invalid operations

TECH STACK:

Frontend

- HTML5
- CSS3
- JavaScript

Backend

- Node.js
- Express.js

Database

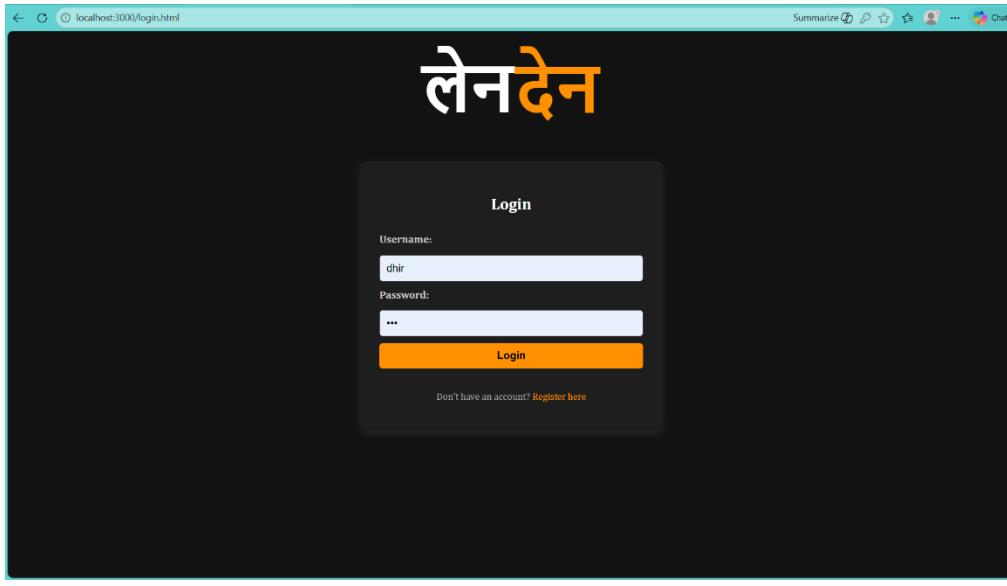
- MongoDB Atlas

Other Tools

- Multer (Image Uploads)
- Express-Session (Authentication)
- Bcrypt (Password Hashing)
- Git & GitHub (Version Control)

Application Screenshots:

Login Page:



Frontend

The login page allows existing users to access the application by entering their username and password. The form is validated on the client side and submitted using JavaScript without reloading the page. Upon successful authentication, the user is redirected to the main buying page, while invalid credentials result in an error message.

Backend Authentication

User authentication is handled on the server using Node.js and Express.js. During registration, user passwords are securely hashed using bcrypt before being stored in the database. During login, the entered password is verified by comparing it with the stored hashed password.

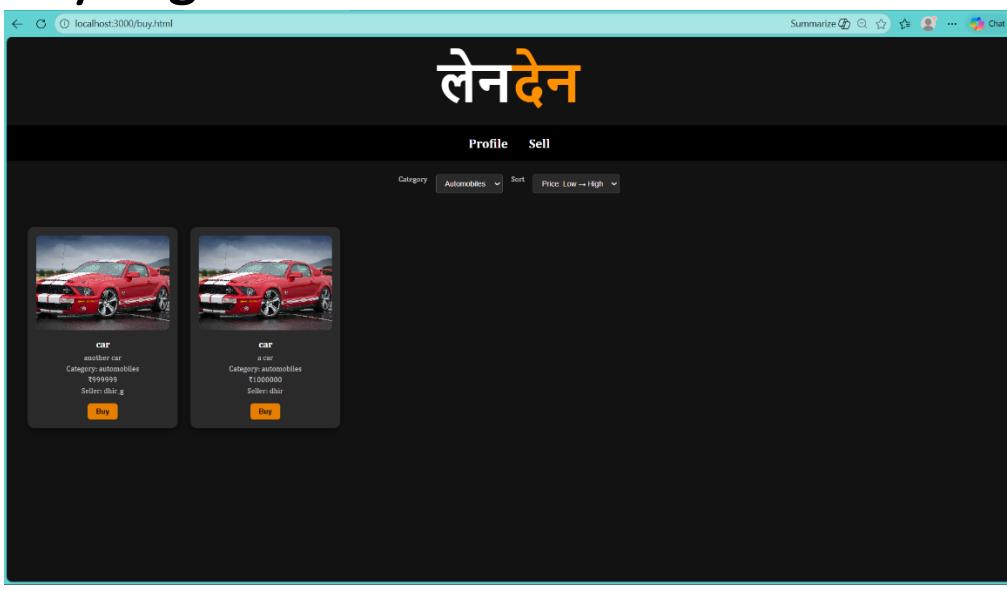
Session Management

The application uses session-based authentication to maintain user login state. A session is created upon successful login and is required to access protected routes such as buying, selling, and profile management. A logout mechanism is provided to securely destroy the session.

Security Measures

- Passwords are stored using bcrypt hashing
- Sessions ensure authenticated access to restricted features
- Unauthorized actions are prevented through server-side checks

Buy Page:



- Displays all items currently available for purchase
- Retrieves item data dynamically from the backend using an API
- Presents items in a responsive grid-based layout
- Each item card includes:
 - Item image
 - Item name and description
 - Category and price
 - Seller information
- Prevents users from purchasing their own listed items
- Provides filtering options based on item categories
- Supports price-based sorting:
 - Low to High
 - High to Low
- Applies filtering and sorting dynamically without page reloads

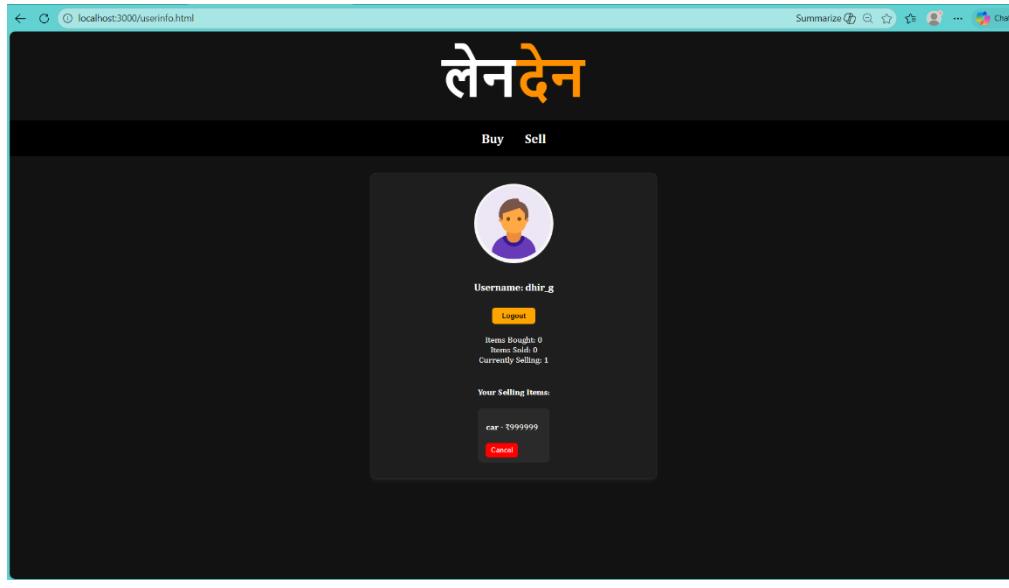
Sell Page:

The screenshot shows a web browser window with the URL `localhost:3000/sell.html`. The page has a dark theme with orange and white text. At the top, there is a logo in Devanagari script and navigation links for 'Buy' and 'Profile'. Below this is a large button labeled 'Sell an Item'. A modal dialog box is open, titled 'Sell an Item'. It contains the following fields:

- Item Name: car
- Description: a car
- Category: Automobiles (selected from a dropdown menu)
- Upload Image: Choose File (with the path `1743998043897-car.jpg` displayed)
- Price (₹): 1000000
- A large orange 'Sell' button at the bottom

- Allows authenticated users to list new items for sale
- Collects item details including:
 - Item name
 - Description
 - Category
 - Price
 - Image upload
- Uses multipart form submission to support image uploads
- Handles image storage on the server using Multer middleware
- Performs basic input validation to ensure all required fields are provided
- Displays a confirmation message after successful item submission
- Adds listed items to the database and makes them visible on the Buy Page

Profile Page:



- Displays personalized information for the authenticated user
- Shows the user's:
 - Username
 - Profile image
 - Activity statistics
- Presents activity statistics including:
 - Number of items bought
 - Number of items sold
 - Number of items currently listed for sale
- Displays a list of items currently being sold by the user
- Allows users to cancel any active listing
- Provides a logout option for secure session termination
- Restricts access to authenticated users only