

Project Report

1. Project Description

The ApplianceRentalHub is an online platform that provides users with easy access to a variety of household and office appliances for temporary rental. This web application addresses the common problem of appliance unavailability in modern apartments and the inconveniences of communal laundry facilities and expensive laundromats. By offering a streamlined rental process, users can easily search for, rent, and manage their appliances without the hassle of long-term commitments.

The platform will also include features for delivery and pick-up options, flexible rental durations, and insurance choices, enhancing user convenience and satisfaction

2. WebApp Functions, Pages, & DB

Technologies:

- Python: For backend business logic to manipulate data and handle request calls.
 - Flask framework – For rendering frontend and routing apis calls
 - pymongo – For Mongo DB connection
- JavaScript: Form validations, Content manipulations, fetch api calls
- HTML5: Adding content structure to the app and design
- CSS3: Styling the whole web-app for User Experience and responsiveness.

Functionalities:

- User Registration & Login: Allow users to create accounts and log in securely.
- Appliance Browsing: Users can search and filter available appliances based on type, brand, and condition.
- Rental Agreement Management: Users can view and manage their rental agreements online.
- Delivery & Pick-Up Scheduling: Users can select their preferred method for receiving appliances.
- Payment Processing: Secure payment processing for rental fees and deposits.
- Maintenance Tracking: Rental agency staff can log maintenance records and view appliance conditions.
- Insurance Options: Users can opt for insurance coverage during the rental process.
- User Dashboard: A personalized area for users to view their rental history and ongoing rentals.

Pages and Files:

- homepage.html: Landing page featuring service overview, login menu, and appliance highlights.
- signup.html & register.js: User registration page to create a new account.
- login.html & login.js: User login page.
- admindashboard.html: Admin interface for managing appliance inventory and maintenance records.
- dashboard.html: User dashboard displaying rental history and account information

- payment.html & payment.js: Secure payment processing page.
- server_main.py: manages all the requests from the frontend and fetches and renders html pages.
- DBManager.py: hands the MongoDB connection and collections
- validator.py: validates the login and signup form data.

3. Application Setup

1. Extract the Zip file: 'web-application.zip'
2. Navigate to the project folder "cd web-application"
3. Install python from the website: <https://www.python.org/>
4. Install required libraries of python my using the below command
pip install -r requirements.txt
5. The Database is connected through MongoDB cloud cluster.
6. Run the service using the command:
python server_main.py
7. Server will be running in the default port 5000 in localhost
8. Check the status by hitting the URL in the browser

<http://localhost:5000>

(Optional)

- I. To setup MongoDB locally
- II. Install MongoDB compass from <https://www.mongodb.com/products/tools/compass>
- III. Create a local connection string such as eg: <mongodb://localhost:27017/> from the MongoDB compass
- IV. Import the collections files (.json) shared in the project zip.
- V. Replace the connection string of the DB in the file *dbManager.py* to access the data from local.

4. Sample DB Data

appliance:

```
{
  "_id": {
    "$oid": "6743c8fe07e7eb9b07cd2999"
  },
  "brand": "BOSCH",
  "model": "TRES-FG",
  "type": "Air Cooler",
  "condition": "New",
  "rental_rate": "55",
  "availability_status": "Available",
  "deposit_amount": "70",
  "serial_number": "QWERT",
  "features": [
    "Instant Cooling"
  ],
  "image_url": "https://www.bosch-industrial.com/ocsmedia/optimized/2000x2000/o522361v389_Air_Flux_6300_v1.png"
}
```

users:

```
{
  "_id": {
    "$oid": "67477aac5d447b1a9dcb470e"
  },
  "firstname": "UserCheck",
}
```

```
    "lastname": "Out",
    "address": "7654W 109ST",
    "email": "uc@gmail.com",
    "phone": "999-999-9999",
    "ssn": "999-99-9999",
    "password": ""
  }
rentalAgreement:
  {
    "_id": {
      "$oid": "674a24d380fde5e3ed6af3f5"
    },
    "appliance_id": {
      "$oid": "6733d509adf627b2e93d4688"
    },
    "customer_id": {
      "$oid": "674751541cf4f83c8fb41593"
    },
    "rental_start_date": {
      "$date": "2024-11-30T00:00:00.000Z"
    },
    "rental_end_date": {
      "$date": "2024-12-07T00:00:00.000Z"
    },
    "quantity": 2,
    "rental_rate": 90,
    "deposit_amount": 180,
    "total_amount": 270,
    "insurance_status": "In Active",
    "return_status": "not returned",
    "damage_report": "none",
    "delivery_type": "delivery"
  }
```

Payment:

```
{
  "_id": {
    "$oid": "674a24d380fde5e3ed6af3f6"
  },
  "agreement_id": {
    "$oid": "674a24d280fde5e3ed6af3f4"
  },
  "amount": 641,
  "payment_date": {
    "$date": "2024-11-29T20:32:19.213Z"
  },
  "status": "completed",
  "card_number": "4242-4242-4242-4242",
  "cvv": "111",
  "expired_date": "11/34",
  "name_on_card": "Hemanth",
  "zip_code": "12345",
  "card_type": "credit"
}
```

customers:

```
{
  "_id": {
```

```
"$oid": "67477af05d447b1a9dcb470f"
},
"customer_id": {
  "$oid": "67477aac5d447b1a9dcb470e"
},
"user_name": "UserCheck Out",
"address": "7654W 109ST",
"phone_number": "999-999-9999",
"email": "uc@gmail.com",
"rental_history": [
  {
    "appliance_id": "6743c8fe07e7eb9b07cd2999",
    "quantity": 5,
    "insurance": "Active"
  },
  {
    "appliance_id": "673feb047c7509e16282a019",
    "quantity": 6,
    "insurance": "InActive"
  }
]
}
```

5. User Interfaces and Forms

1. Login page – login.html

← → ↻ 📑 🌐 🏠

localhost:5000/login.html

☆ 📄 🗑️ 🌐 ⋮

Login

Username:

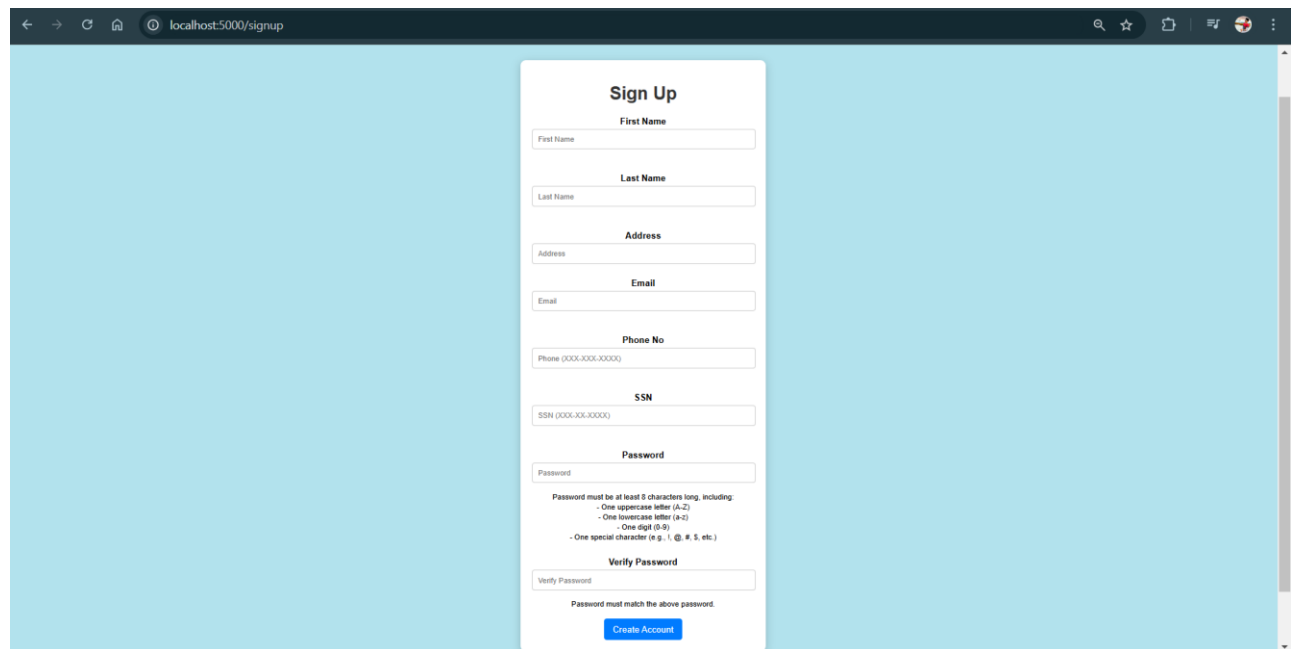
Password:

[Forgot password?](#)

Login

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

2. User Registration Page – Signup.html



Sign Up

First Name

Last Name

Address

Email

Phone No

SSN

Password

Verify Password

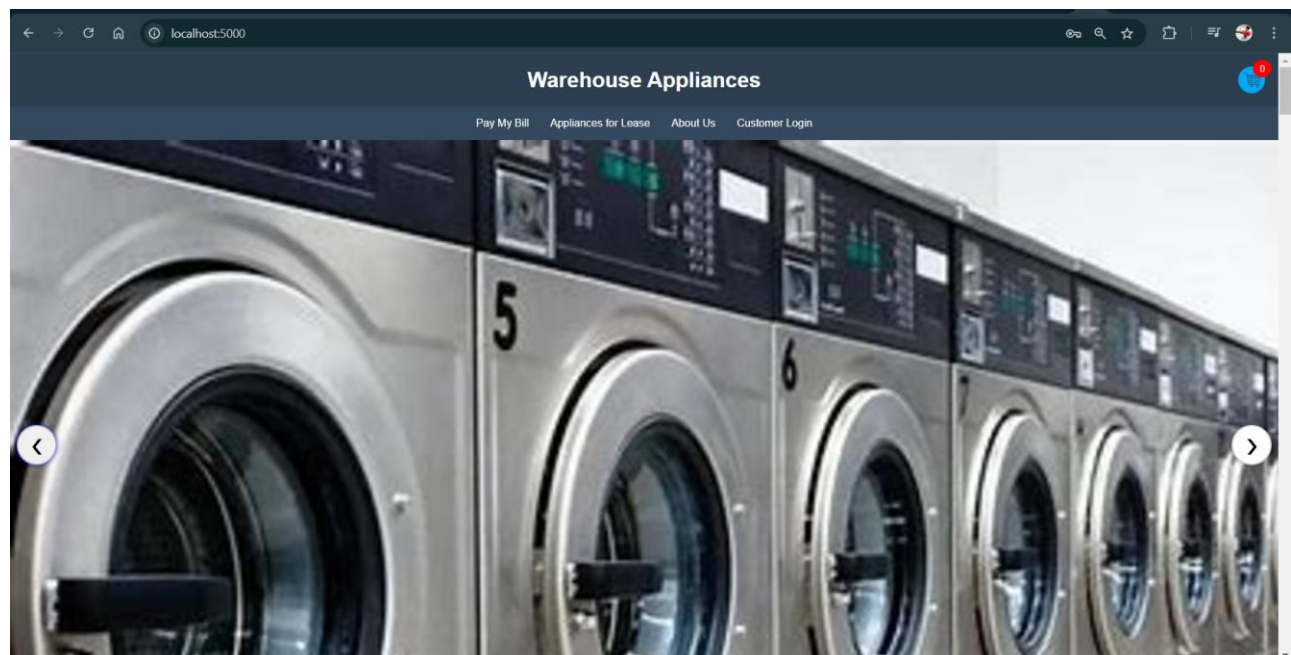
Password must be at least 8 characters long, including:

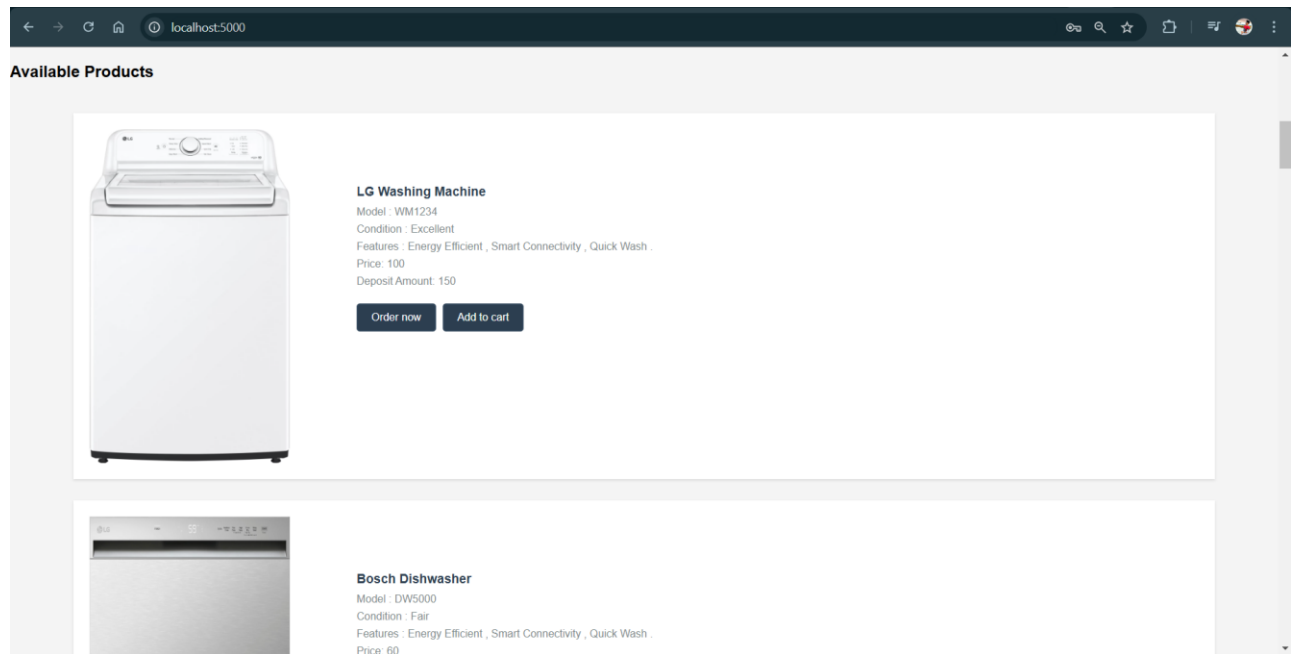
- One uppercase letter (A-Z)
- One lowercase letter (a-z)
- One digit (0-9)
- One special character (e.g., !, @, #, \$, etc.)

Password must match the above password.

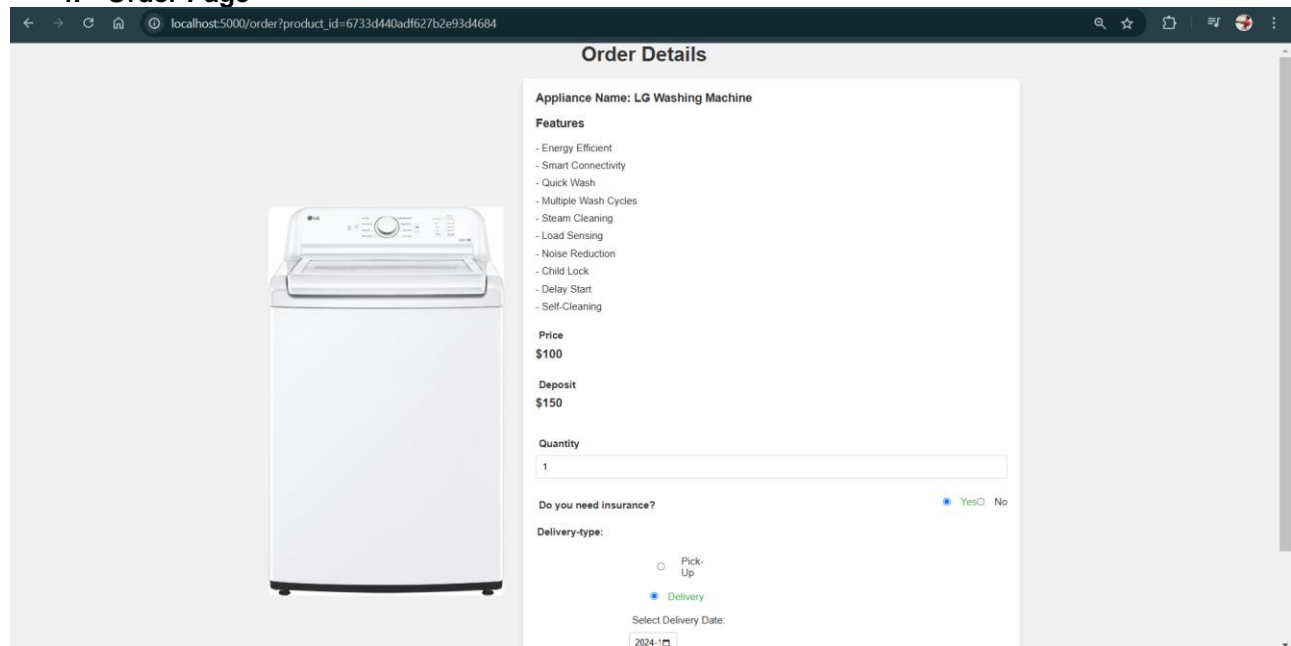
[Create Account](#)

3. HomePage – homepage.html

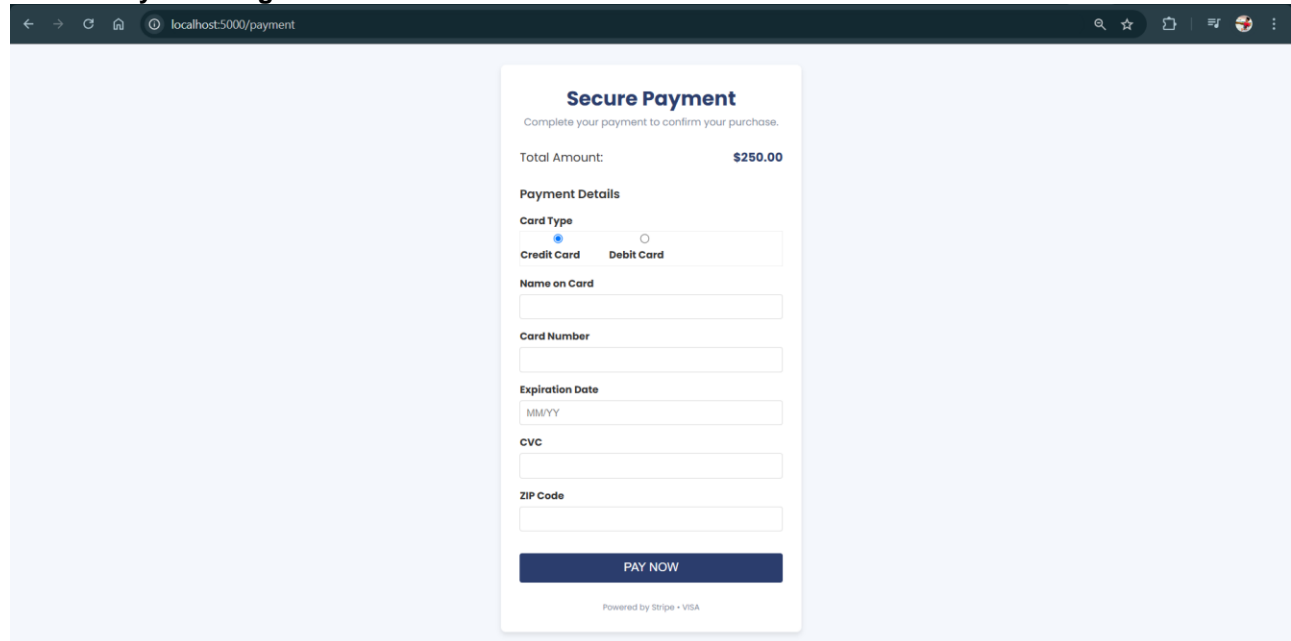




4. Order Page



5. Payment Page



A screenshot of a web browser showing a payment page. The browser's address bar displays 'localhost:5000/payment'. The page features a central white card with a blue header 'Secure Payment' and a sub-header 'Complete your payment to confirm your purchase.' Below this, the 'Total Amount' is listed as '\$250.00'. The 'Payment Details' section includes a 'Card Type' selector with 'Credit Card' selected and 'Debit Card' as an option. There are input fields for 'Name on Card', 'Card Number', 'Expiration Date' (with a 'MM/YY' placeholder), 'CVC', and 'ZIP Code'. A prominent blue 'PAY NOW' button is at the bottom of the card, with a small note 'Powered by Stripe - VISA' underneath it.

Secure Payment
Complete your payment to confirm your purchase.

Total Amount: **\$250.00**

Payment Details

Card Type
☒ Credit Card ☐ Debit Card

Name on Card

Card Number

Expiration Date

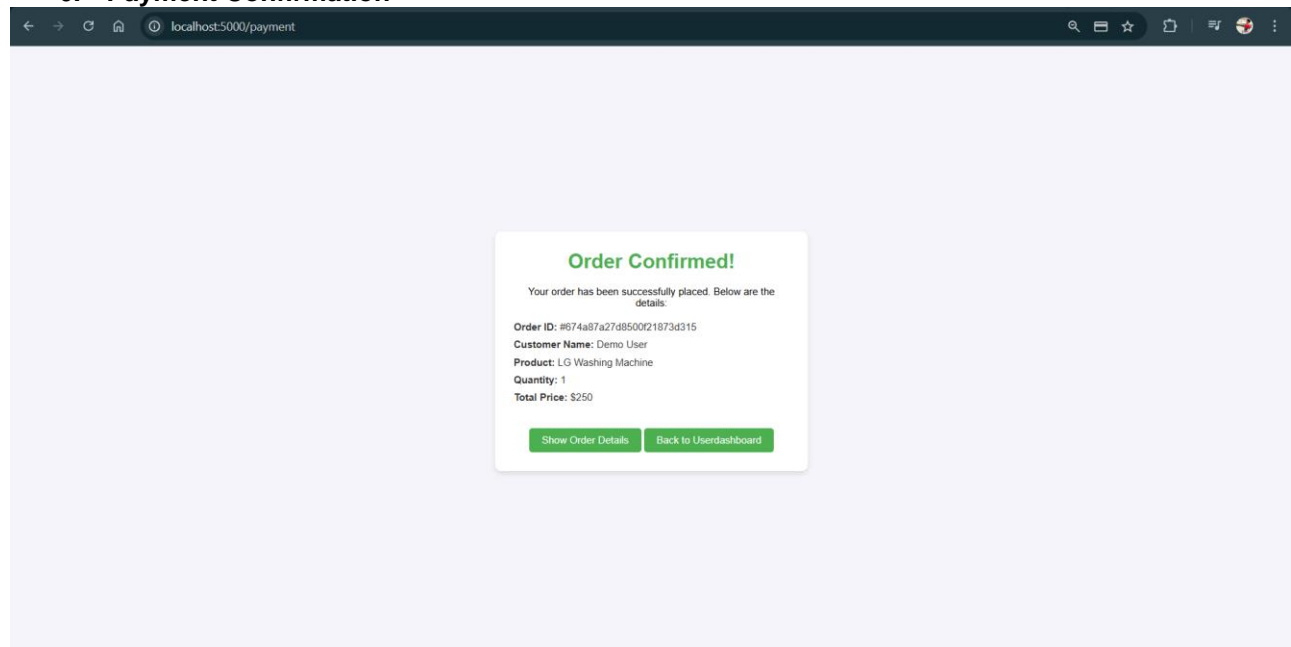
CVC

ZIP Code

PAY NOW

Powered by Stripe - VISA

6. Payment Confirmation



A screenshot of a web browser showing a payment confirmation page. The browser's address bar displays 'localhost:5000/payment'. The page features a central white card with a green header 'Order Confirmed!'. Below the header, a message states 'Your order has been successfully placed. Below are the details:'. The details listed are: 'Order ID: #674a87a27d850021873d315', 'Customer Name: Demo User', 'Product: LG Washing Machine', 'Quantity: 1', and 'Total Price: \$250'. At the bottom of the card, there are two green buttons: 'Show Order Details' and 'Back to Userdashboard'.

Order Confirmed!

Your order has been successfully placed. Below are the details:

Order ID: #674a87a27d850021873d315
Customer Name: Demo User
Product: LG Washing Machine
Quantity: 1
Total Price: \$250


[Show Order Details](#) [Back to Userdashboard](#)

7. User Dashboard

← → ↻ 🏠 🌐 localhost:5000/dashboard 🔍 📁 ☆ 🗑️ 🖨️ 🌐 ⋮

☰ Menu Welcome, Demo D

Ongoing Rentals




Daikin FTKF51 - (Air Conditioner)

Start Date: 2024-11-27

End Date: 2024-12-04

Return Status: not returned




Panasonic NN-SN966S - (Microwave)

Start Date: 2024-11-27

End Date: 2024-12-04

Return Status: not returned




BOSCH TRES-FG - (Air Cooler)

Start Date: 2024-11-28

End Date: 2024-12-05

Return Status: not returned




Panasonic NN-SN966S - (Microwave)

Start Date: 2024-11-30

End Date: 2024-12-07

Return Status: not returned




LG WM1234 - (Washing Machine)

Start Date: 2024-11-28

End Date: 2024-12-07

Return Status: not returned




Daikin FTKF51 - (Air Conditioner)

Start Date: 2024-11-28

End Date: 2024-12-05

Return Status: not returned




Panasonic NN-SN966S - (Microwave)

Start Date: 2024-11-30

End Date: 2024-12-07


Return Status: not returned



Daikin FTKF51 - (Air Conditioner)

Start Date: 2024-11-28

Rental History



Panasonic NN-SN966S - (Microwave)

Start Date: 2024-11-30

End Date: 2024-12-07

Return Status: returned


8. Admin Dashboard

← → ↻ 🏠 🌐 localhost:5000/admindashboard 🔍 📁 ☆ 🗑️ 🖨️ 🌐 ⋮

☰ Menu Welcome, admin A

Appliance List

Search by name... Add Appliance



Brand: LG

Type: Washing Machine

Rental Rate: \$ 100

Deposit Amount: \$ 150

Features: Energy Efficient, Smart Connectivity, Quick Wash, Multiple Wash Cycles, Steam Cleaning, Load Sensing, Noise Reduction, Child Lock, Delay Start, Self-Cleaning

Model: WM1234


Condition: Excellent

Availability: Out of Stock

Serial Number: SNG54321

Edit

Delete



Brand: Bosch

Type: Dishwasher

Rental Rate: \$ 60

Deposit Amount: \$ 120

Features: Energy Efficient, Smart Connectivity, Quick Wash, Multiple Wash Cycles, Steam Cleaning, Load Sensing, Noise Reduction, Child Lock, Delay Start, Self-Cleaning

Model: DW5000


Condition: Fair

Availability: Available

Serial Number: SNG98765

Edit

Delete



Brand: Panasonic

Type: Microwave

Rental Rate: \$ 45

Deposit Amount: \$ 90

Features: Inverter technology, 10 power levels, Auto-cook options, Child lock

Model: NN-SN966S


Condition: New

Availability: Available

Serial Number: SNG112233

Edit

Delete



Brand: Daikin

Type: Air Conditioner

Rental Rate: \$ 120

Deposit Amount: \$ 251

Features: Energy-efficient Inverter technology, Auto mode, Turbo cooling, Remote control operation, Air purification filter, Overload protection, Auto shut-off on overheating, Child locking feature

Model: FTKF51

Condition: Used

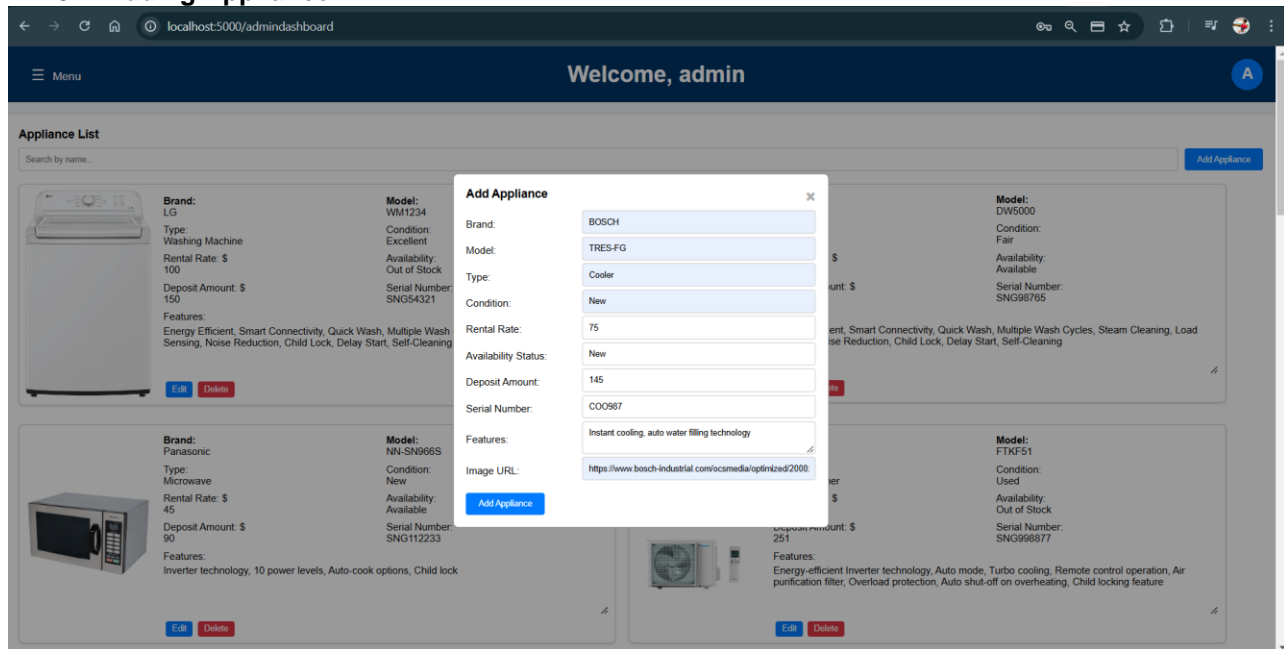
Availability: Out of Stock

Serial Number: SNG998877

Edit

Delete

9. Adding Appliance



6. Source Code

<https://github.com/KOTHAPALLIRAJASEKHARREDDY/Web-application>

7. Live Application Link

<http://web-application-tawny.vercel.app/>

8. Contributions:

The project team will be divided into roles based on strengths and interests.

- **Student #1: Sai Kumar Pagalla**
 - Designing the database schema and manages all SQL queries for data retrieval and manipulation.
 - Will utilize HTML, CSS, and JavaScript for responsive design and user experience enhancements for login page.
- **Student #2: Rajasekhar Reddy Kothapalli**
 - Handling server-side programming, focusing on python flask framework for data processing, session management, and API routing.
 - Connecting MongoDB cloud cluster through pymongo and designing DBLayer for database operations.
 - Responsible for designing homepage, signup validations, order page routing post login, payment, and confirmation page design.
- **Student #3: Hemanth Rasabhathula**
 - Designing the User and Admin Dashboards, Cart page and data processing for backend api calls to make CRUD operations with MongoDB.
 - Responsible for storing data in local Storage and making the cart functionality persist across user session and go through payment module.

Conclusion

The ApplianceRentalHub is a rental Application which uses python for backend data logic and the JavaScript, HTML & CSS along with form validations and css manipulations to give a better user experience and intuitive visibility through out the application. For DB MongoDB is used with relational Normalization between collections and stores and retrieved using pymongo library provided by python.