

Blinkit Clone ECommerce - project

The project is Web app frontend part as well as backend part. In side project work on frontend part. Use the NodeJS, React, Bootstrap, HTML, CSS etc. and backend side used CORS, mongoose, express, and so on. This project creating for ecommerce. In This project home, about us, contact and cart summary are present. This app created by using React JS application based and design is fully responsive.

1. Frontend: -

It's a created using the react application. Inside react application installed react packages. In-between bootstrap library installed.

In Application created Nav Bar section created in between Nav Bar section Added some pages Like Login page, signup page, Home page, About page, menu page, and contact page. The Nav Bar is clickable to rendering one page to another page.

This Application is fully responsive created.

web App previews is below:

NATURAL FARM FRUITS

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo

[SHOP NOW](#)


Fresh Fruits



Apple
fruits
₹20

[Add Cart](#)


Apples
fruits
₹100

[Add Cart](#)


Banana
fruits
₹50

[Add Cart](#)


Black-Grapes
fruits
₹100

[Add Cart](#)


Dragon-Fruit
fruits
₹80

[Add Cart](#)


Fruits
fruits
₹500

[Add Cart](#)

Your Product





Fruits



Vegetable



Apple
fruits
₹20

Add Cart



Apples
fruits
₹100

Add Cart



Banana
fruits
₹50

Add Cart



Black-Grapes
fruits
₹100

Add Cart



Dragon-Fruit
fruits
₹80

Add Cart



Fruits
fruits
₹500

Add Cart



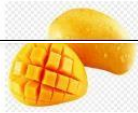
Guava
fruits
₹70

Add Cart



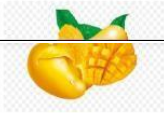
Kiwi
fruits
₹30

Add Cart



Mango
fruits
₹30

Add Cart



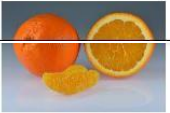
Mangoes
fruits
₹150

Add Cart



Musk-Melon
fruits
₹60

Add Cart



Orange
fruits
₹40

Add Cart



Pine-Apple
fruits
₹50

Add Cart



Pine-Apples
fruits
₹200

Add Cart



Raw Mango
fruits
₹40

Add Cart



Watermelon
fruits
₹100

Add Cart



Amaranthus
vegetable
₹40

Add Cart



Beetroot
vegetable
₹30

Add Cart

localhost:3000/menu/65fc5a2b4e706059fa979001



Brinjals
vegetable
₹50

Add Cart



Cabbage
vegetable
₹20

Add Cart



Capsicum
vegetable
₹20

Add Cart



Cauliflower
vegetable
₹50

Add Cart



Coriander
vegetable
₹10

Add Cart



Cowpea Beans
vegetable
₹40

Add Cart



Garlic
vegetable
₹50

Add Cart

Ginger
vegetable
₹35

Add Cart

Green-Peas
vegetable
₹15

Add Cart

Onion
vegetable
₹25


Add Cart

Peeled-Garlic
vegetable
₹50

Add Cart


Potatoes
vegetable
₹20

Add Cart




Red-Capsicum
vegetable
₹10

Add Cart



Red-Carrot
vegetable
₹50


Add Cart



Spring-Onion
vegetable
₹30

Add Cart





localhost:3000/menu/65fc5db34e706059fa979023



India's Last Minute App

ABOUT SHOP

[About](#)
[Privacy Policy](#)
[Contact](#)



Customer support

+91-85580-30604

Mon. - Fri. 10:00 - 7:00
Sat. - Sun. Off

© 2024 BlinkIt. All Rights Reserved.

SignUp Page



First Name

Last Name

Email

Password

Confirm Password

Sign up

Already have an account ? [Login](#)

Login Page

Email

shekharsuri25@gmail.com

Password

.....



Login

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

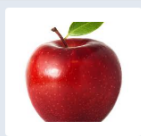
blinkit

India's Last Minute App

[Home](#) [About](#) [Contact Us](#)



My Cart



Apple

fruits

₹20

+ 1 -

Total : ₹20



Apples

fruits

₹100

+ 1 -

Total : ₹100



Banana

fruits

₹50

+ 1 -

Total : ₹50

Summary

Total Qty : 3

Total Price ₹ 170

Payment

We're Here

We believe BlinkIt Product it's the heart of every customer

Address

Office #13, NSTP, Bangalore, India

Email  info@blinkit.com

Let's Talk

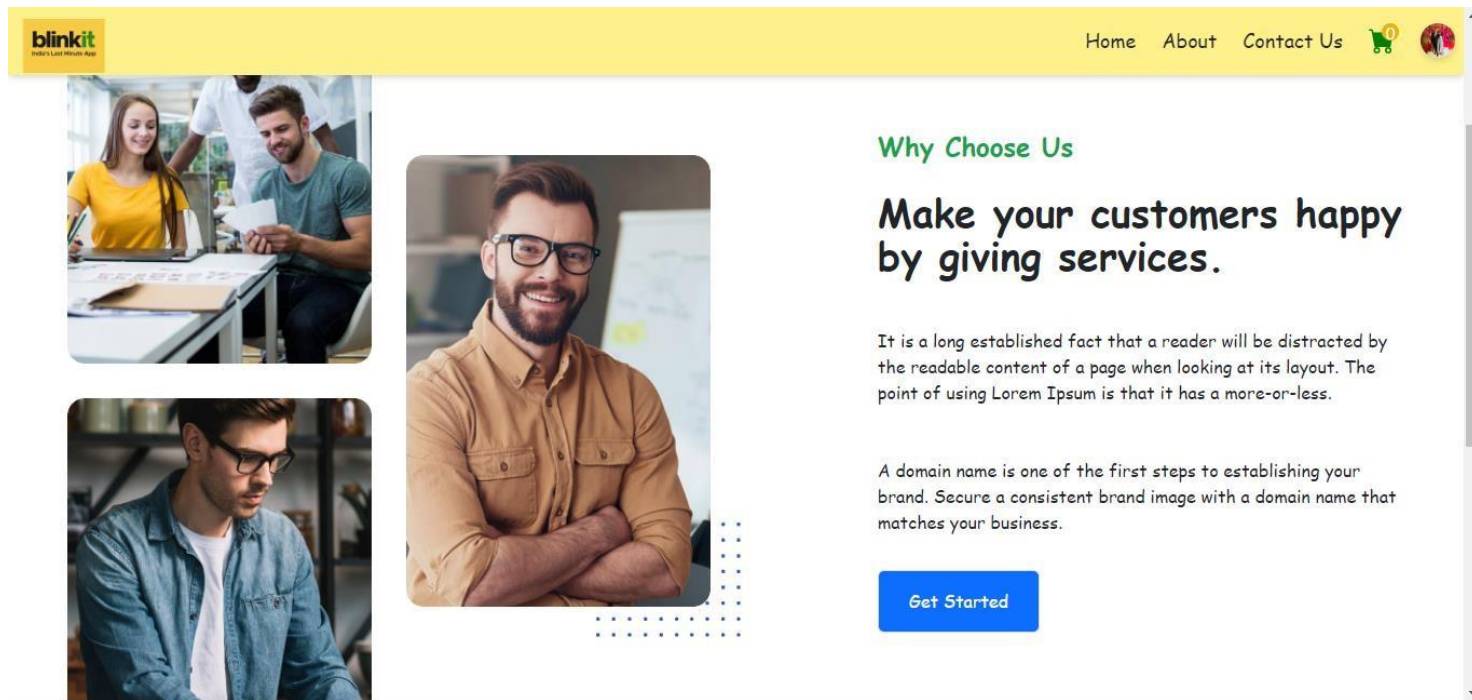
For enquiries, please email us using the form below

Name

Email Address

Message

Send

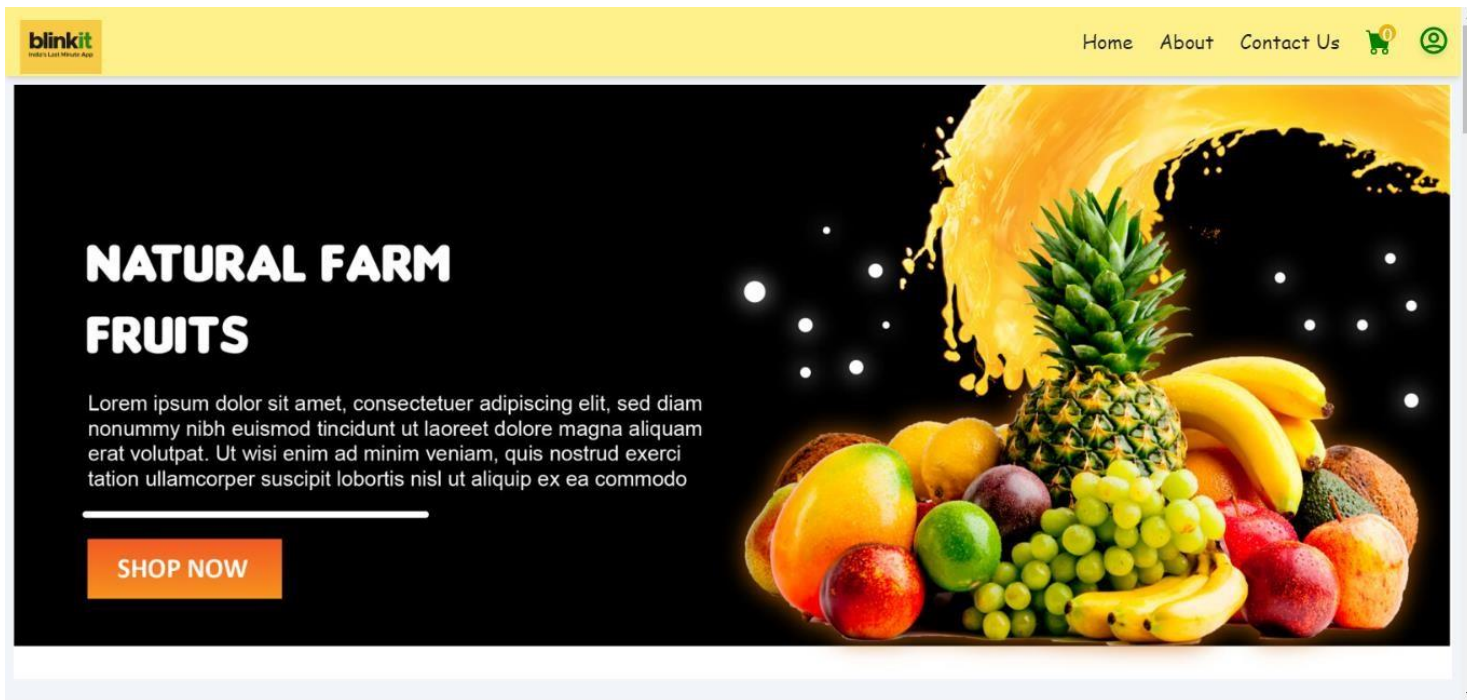


- In side two type login present

1. Admin Logs:

These logs capture actions or events related to administrative activities. They may include operations like Add new product, user management, or any activity that involves administrative privileges.

Admin log panel :



In side right top corner New product option only present of admin By through Admin added the new product. Those new product added on the application.

New product click then open new product added page look below:

The image shows a form for adding a new product. The form is centered on a light blue background. It has a white header with the 'blinkit' logo on the left and navigation links 'Home', 'About', and 'Contact Us' on the right, along with a shopping cart icon and a user profile icon. The form itself is a white box with a light blue border. It contains the following fields: 'Name' (a text input field), 'Category' (a dropdown menu with 'select category' as the placeholder text), 'Image' (a large text input field with a cloud upload icon in the center), 'Price' (a text input field), and 'Description' (a text input field). At the bottom of the form is a yellow button with the text 'Save'.

Inside Product input, category option, image input, price, and description present after all information add then after save click they product added to the selected category inside stored.

2. User Logs:

These logs capture actions or events performed by regular users. They include activities related to the application's core functionality, such as data entry, updates, or other userspecific interactions.

User login not show this option they only product see, add cart, buy product this option available.

2. Backend: -

Inside providing a server-side code writing in Node.js using Express, which server as an API for task management application.

I Express Setup:

- The express package is used to create an Express application.
- The CORS package is used to enable Cross-Origin Resource Sharing, allowing your frontend to make requests to the server.

II MongoDB Atlas Storage:

- MongoDB Atlas is a fully-managed cloud databases service for MongoDB.
- It provides a convenient way to deploy, operate, and scale MongoDB databases in the cloud.

III Middleware:

- The `express.json()` middleware is commented out but can be uncommented to enable JSON body parsing.

IV Error Handling Middleware:

- There's a centralized error-handling middleware that catches errors and sends a 500 Internal Server Error response.

V Endpoints:

- GET /tasks: Retrieves all tasks stored in the server's storage.

- **POST /tasks:** Adds a new task to the storage. It expects a JSON payload with a text property.
- **DELETE /tasks/:taskId:** Deletes a task with the specified ID.
- **PUT /tasks/:taskId:** Updates a task with the specified ID. It expects a JSON payload with a text property.

VI Server Initialization:

- The server listens on port 8800, and a message is logged to the console upon successful startup.

In this React application you are directly open the home page with login or without login. User directly items watch or added the product in product But without login not purchase this product.

3. Database:

MongoDB Overview:

- **Type:** MongoDB is a NoSQL database management system.
- **Data Model:** MongoDB stores data in BSON (Binary JSON) format, which is a binary-encoded serialization of JSON-like documents.
- **Schema-less:** MongoDB is schema-less, meaning you can insert records without first defining the structure, unlike traditional relational databases.

MongoDB is a NoSQL database that is widely used for its flexibility and scalability.

MongoDB Atlas:

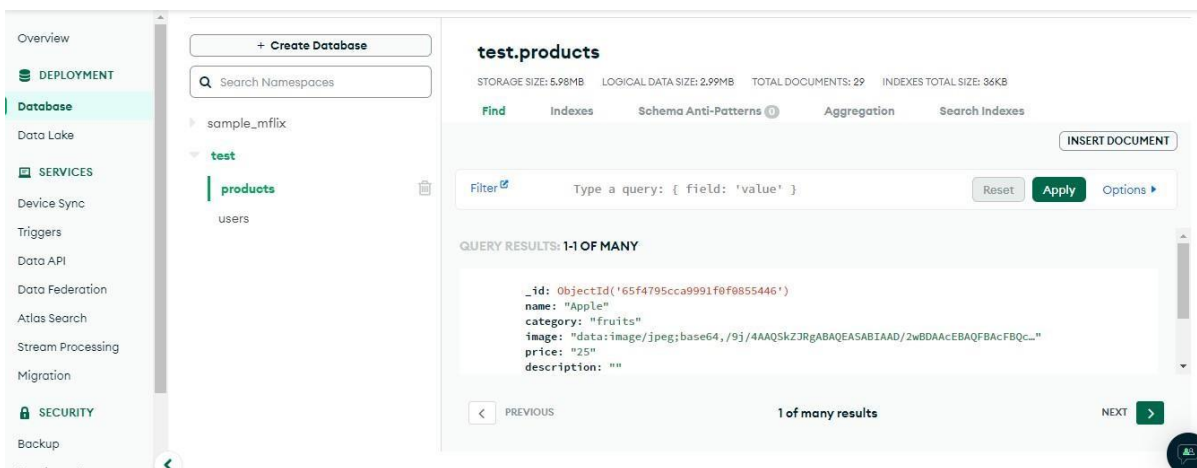
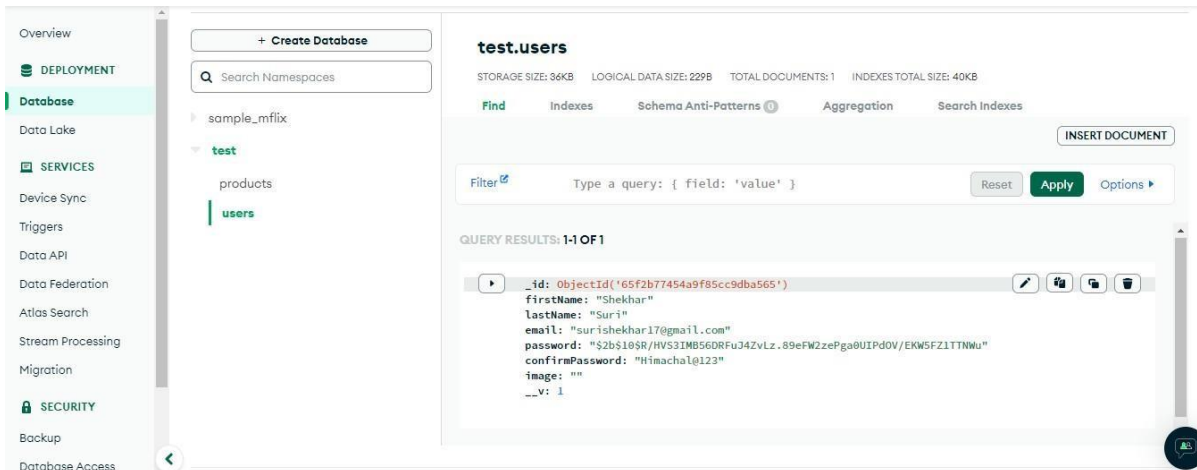
If you are using MongoDB Atlas (cloud-based MongoDB service):

- **Scalability:** Easily scale your MongoDB database as your application grows.
- **Security Features:** Atlas provides security features like network isolation, encryption, and automated backups.
- **Monitoring and Alerts:** Get insights into your MongoDB performance with monitoring and receive alerts.

Security:

- **Authentication:** MongoDB supports various authentication mechanisms.
- **Authorization:** Role-based access control.

In MongoDB created one database. Database name is test. In side two collection created one for product and second for user. Product collection is used for the product data stored. And User collection are used for collect the all information of user.



NOTE:

Before running project run the command of **npm install** in Frontend Terminal as well as backend terminal for install node modules.

Frontend Terminal used **npm start** command to run frontend server.

And backend run **node server.js** command to run backend server.

