

# EVERYDAY MART

## MID – TERM REPORT OF MINI PROJECT - II

### BACHELOR OF TECHNOLOGY Computer Science and Engineering Branch

#### SUBMITTED BY:

ABHINAV BHARDWAJ (181500009)  
ADITYA SINGH CHAUHAN (181500046)  
ANKIT PARMAR (181500099)  
ANVIT GUPTA (181500127)  
SOMIYA PARMAR (181500718)

#### SUPERVISED BY:

Mr. NEERAJ KHANNA  
( Technical Trainer)



**GLA University, Mathura**  
**(2020 – 2021)**

---

# Table of Content

Abstract.....	3
Introduction .....	3
Problem Definition .....	3
OBJECTIVE.....	4
Technology Used.....	4
Methodology.....	5
Implementation Details .....	5
Webpage Based Portal: .....	5
Frontend Development: .....	5
Backend Development: .....	6
Android Based Application: .....	6
Frontend Development: .....	6
Progress Details .....	7
On Webpage Based Portal:.....	7
Backend: .....	7
On Android Based Application: .....	8
Android Based Application Screenshots.....	9
Screenshots of Web Page Based Portal.....	12
Backend Screenshot: .....	12
Successfully linked Project with MongoDB: - .....	12
Successful Signup using Postman: - .....	13
If tried to sign up with duplicate data that is already in database, no entry in database.....	16
If tried to sign up with invalid mobile number, no entry in database.....	17
If tried to sign up with invalid email address, no entry in database .....	18
And, again signup with proper details.....	19
Successful signed in with correct credentials.....	20
Signing out .....	21
Trying to sign in with incorrect email address that does not exist in database .....	21
Frontend Screenshots: .....	23
Home Page: .....	23
Categories section: .....	23
Product Section: .....	24
New Product Section: .....	25
Offer Section:.....	25

---

Footer Section: .....26

All Product Section: .....26

Individual Product Detail: .....27

## **Abstract**

In this era of internet, e-commerce is growing by leaps and bounds keeping the growth of brick-and-mortar businesses in the dust. In many cases, brick-and-mortar businesses are resorting to having a counterpart which is internet or e-commerce driven. People in the developed world and a growing number of people in the developing world now use e-commerce websites on a daily basis to make their everyday purchases. Still the proliferation of e-commerce in the under-developed world is not that great and there is a lot to desire for.

## **Introduction**

The busy schedule usually ceases us from visiting market for buying essential commodities. Most of the shopping stores are open during day time only and we do not have enough time during the day to go out and shop because of other commitments. Also, overfilled stores never create a pleasant experience for the people who shop. Nobody would ever like to spend their precious time while standing on queues.

Our system will provide facility to the customer of the shop, to shop online using the website and mobile app of the shop. It will prove to be of great use for the shopkeeper as well, as it will provide effectual and worthful interaction with the customers. It will also be very useful during the tough times like pandemic. Providing customer to shop online from the shop of their choice, getting their essentials at home without going outside, and providing the shopkeeper to interact with their customers, thus, providing better customer service and hence, facilizing and improving customer relations, our system will prove to be of great importance to the people.

## **Problem Definition**

Offline shopping is a physical activity that takes too much time and offline billing systems require waiting in long queues. The rush created by offline stores due to crowds sometimes goes beyond expectations and becomes too time consuming. The hectic schedule of the people prevents them from visiting offline stores and thus purchasing necessary stuff also gets affected. All this creates an urgent need for an alternative solution which will facilitate customers as well as shopkeepers to keep in touch with each other. Our system will offer a website and mobile application for an offline store to function online as well.

# OBJECTIVE

Everyday Mart is an android as well as web application where users can purchase and order essentials online. The system is developed with a user-friendly and attractive GUI. It delivers a wide range of essentials available online. Users have to first login into the system to view the essentials and add them into their cart. They can then order it by making a secure online payment. The system functionality of products and orders is stored on server side in a web service. The android app is for client usage.

It consists of client-side scripting for placing orders by connecting to the server-side web service.

- Reduce management costs
- Developing business relations
- Providing a unique customer experience
- Developing relevant target
- Increasing sales

## Technology Used

### Hardware Requirements:

- Computer System with minimum 8GB of RAM

### Software Requirements:

- Windows/Linux OS
- Android Studio
- Visual Code Studio
- Robo 3T
- Postman
- Adobe XD

## Programming language, Framework and Libraries:

- Java Programming
- JavaScript
- HTML
- CSS
- ReactJ
- Express
- NodeJS

## Methodology

We are using Android Java and React based Webpage Portal as the frontend with the backend made using MongoDB. To see what's inside the android application a user has to login, if the user is not register, he/she can also register himself/herself. App will keep records of previous order of the customer and the feedback.

The modules used in this app are listed as follows:

- Main Dashboard
- My account
- add to cart
- Order records
- Payment
- Logout

## Implementation Details

The implementation is divided in two parts:

### Webpage Based Portal:

**Frontend Development:** We will be using React, HTML, Bootstrap for frontend development part. On home page, there will be all the product listed, major categories options, new products list and offer section. There is further option of home, products, contact us and about us. User

can create account by Account Page. After signing in into account, there will be various modules like My Cart, Place Order, Order Detail, Sign-out, etc.

**Backend Development:** The project commences with designing of fundamental user schema and few other schemas. We will use APIS for creating different routes. We will differentiate user as customer and admin (seller) with the help of middleware. Admin will be able to use functionalities related to managing products like adding new categories, adding products to that category, etc. The data will be sent to backend using APIs.

An application program interface (API) is a set of routines, protocols and tools for building software applications. Basically, an API specifies how software components should interact.

The data will be sent in JSON format.

JSON is a lightweight data-interchange format it is easy for humans to read and write. It is easy for machines to parse and generate.

We are using MongoDB Atlas for cloud storage during development phase because of its flexibility and scalability of document database, available as a fully managed service.

MongoDB Atlas is the global cloud database service for modern applications. Deploy fully managed MongoDB across AWS, Azure, or GCP. Best-in-class automation and proven practices guarantee availability, scalability, and compliance with the most demanding data security and privacy standards.

## Android Based Application:

**Frontend Development:** We are using android studio, the tool which is used to develop android applications. The technology used for developing frontend is Android and Java. User will have to register themselves before login. We are parsing the data coming in JSON from the backend and displaying it on android application.

The modules used in this app are listed as follows:

- 1) Login Page - will use to sign in to the account.
- 2) Sign Up – will use to create Seller as well as Customer Account
- 3) Edit Profile- In this Both can Edit their details.
- 4) From Seller Ends:
  - (i) Add or delete product- In this Seller can add or delete their products according to their availability.

- (ii) Show Products- By this seller can make products to visible or not.
  - (iii) Monitor Orders – In this seller can keep records of their Orders.
  - (iv) Order Details- Can keep track on orders details.
  - (v)
- 5) From Customer Ends:
- (i) Watch Shop & Shop Details
  - (ii) Add to cart: By this customer can add product to his cart for buying.
  - (iii) Show cart: In this customer can watch the item he/she added to the cart.
  - (iv) Place Order: In this He can final his order.
- 6) Notification Of order: In this customer can track order.

## **Progress Details**

### **On Webpage Based Portal:**

#### **Backend:**

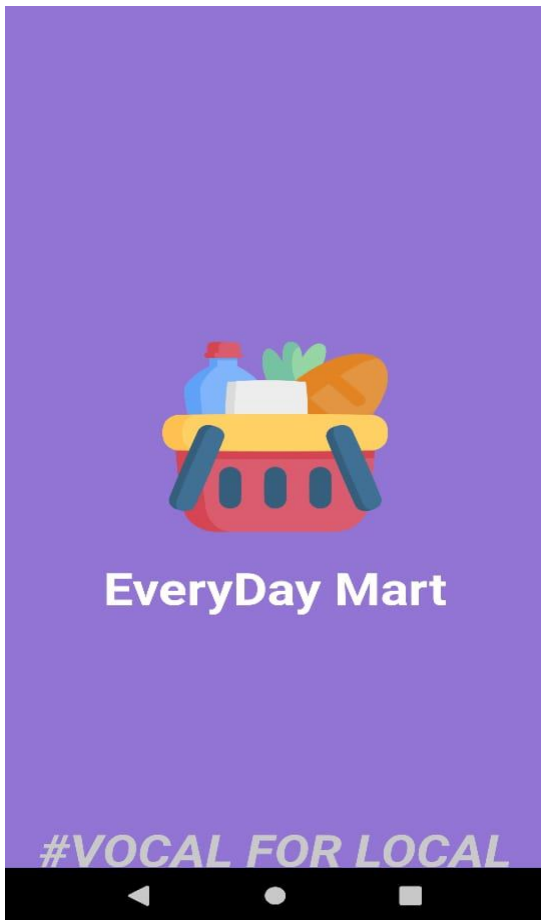
- Created Database on MongoDB, so that it can be used for Web Page based Portal.
- Successfully connected Backend of the Web Page based Portal with database.
- Created a basic Schema for User.
- Created all the controllers required for creating a new account of the User.
- Created all the controllers required for signing in account of the User.
- Created all the controllers required for signing out User from his/her account.
- Created the route required for creating new account of the User as  
http://localhost:XXXXX /api/signup.
- Created the route required for signing in account of the User as http://localhost:XXXXX  
/api/sign in.
- Created the route required for signing out Patient (User) from his/her account as  
http://localhost:XXXXX /api/sign-out.



## **On Android Based Application:**

- Created app using Android studio, so that it can be used for GUI based Mobile application.
- Created Home page / introductory page.
- Created Sing-in /login page through which User/Seller can access their account.
- Created Sing-up for Customer /register page through which Customer can register in the account by basic details.
- Created Sing-up for Seller /register page through which Seller can register in the account by basic details.

# Android Based Application Screenshots

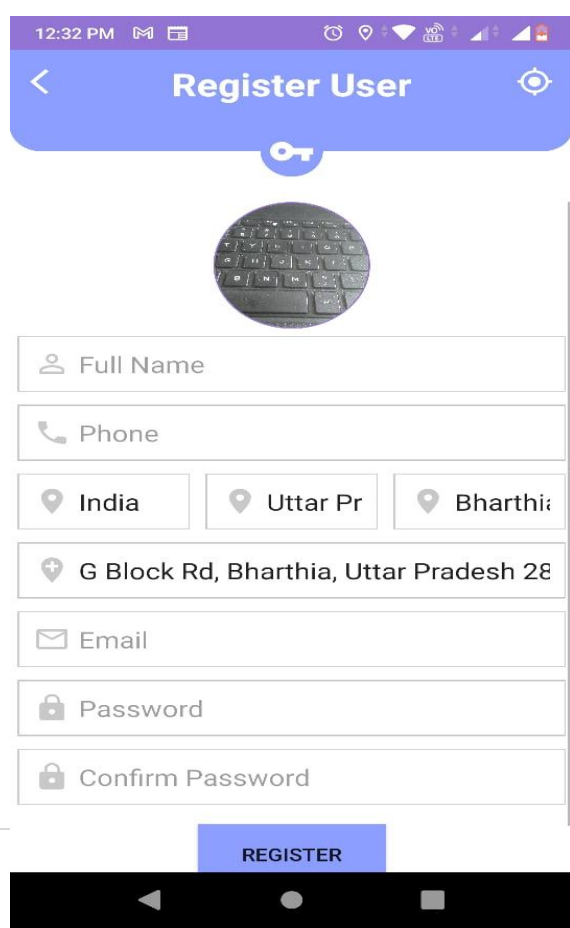
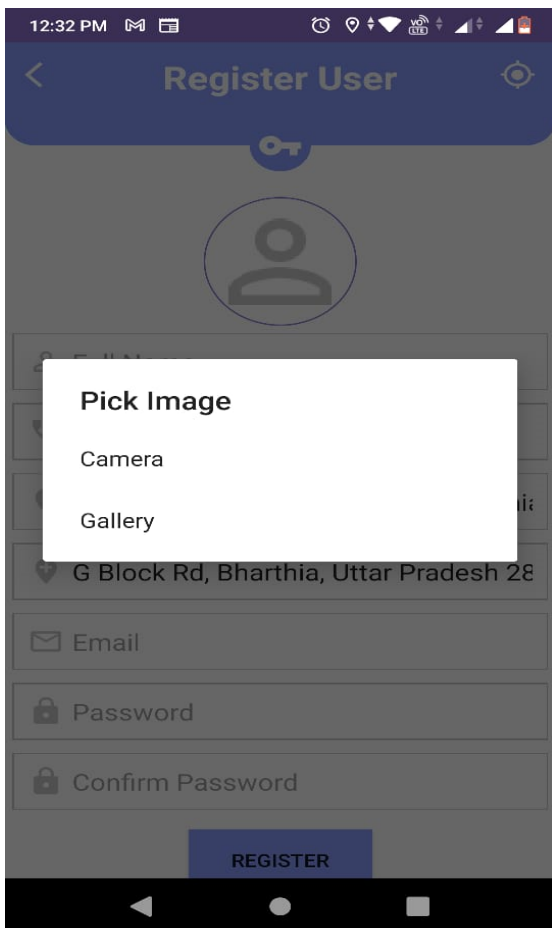
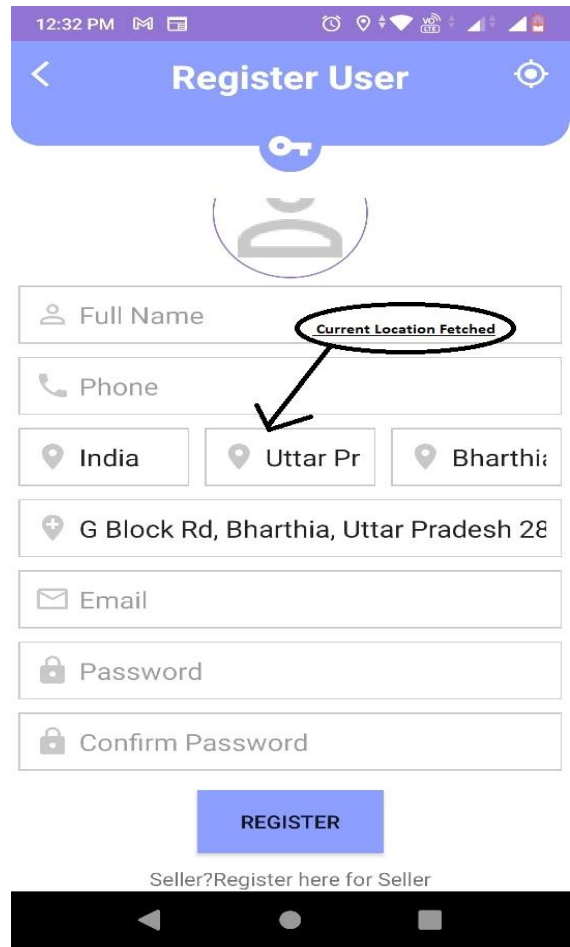
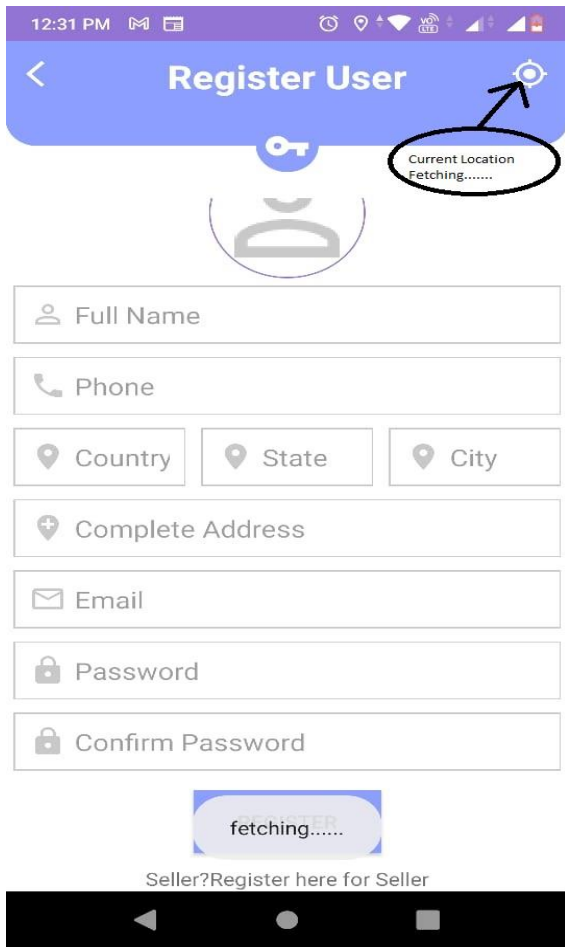


The login screen has a blue header with the word "Login" in white. Below the header is a white key icon. The main area contains two white input fields: "Email" with an envelope icon and "Password" with a lock icon. To the right of the password field is a link that says "Forgot Password?". At the bottom center is a blue button labeled "LOGIN". The Android navigation bar is at the bottom.

Don't have Account? Register here

The register screen for users has a blue header with a back arrow, the text "Register User", and a key icon. Below the header is a circular profile icon placeholder. The form consists of several white input fields: "Full Name" with a person icon, "Phone" with a phone icon, "Country" with a location pin icon, "State" with a location pin icon, "City" with a location pin icon, "Complete Address" with a location pin icon, "Email" with an envelope icon, "Password" with a lock icon, and "Confirm Password" with a lock icon. At the bottom center is a blue button labeled "REGISTER". The Android navigation bar is at the bottom.

The register screen for sellers has a blue header with a back arrow, the text "Register User", and a key icon. Below the header is a circular profile icon placeholder. The form includes white input fields for "Full Name" (with a person icon), "Phone" (with a phone icon), "Country" (with a location pin icon), "State" (with a location pin icon), "City" (with a location pin icon), "Complete Address" (with a location pin icon), "Email" (with an envelope icon), "Password" (with a lock icon), and "Confirm Password" (with a lock icon). At the bottom center is a blue button labeled "REGISTER". Below the button is a link that says "Seller? Register here for Seller". The Android navigation bar is at the bottom.



12:32 PM

## Register Seller

Full Name

Shop Name

Phone

Delivery Fee

Country State City

Complete Address

Email

Password

Confirm Password

REGISTER

12:32 PM

## Register Seller

Profile Icon

Full Name

Shop Name

Phone

Delivery Fee

Country State City

Complete Address

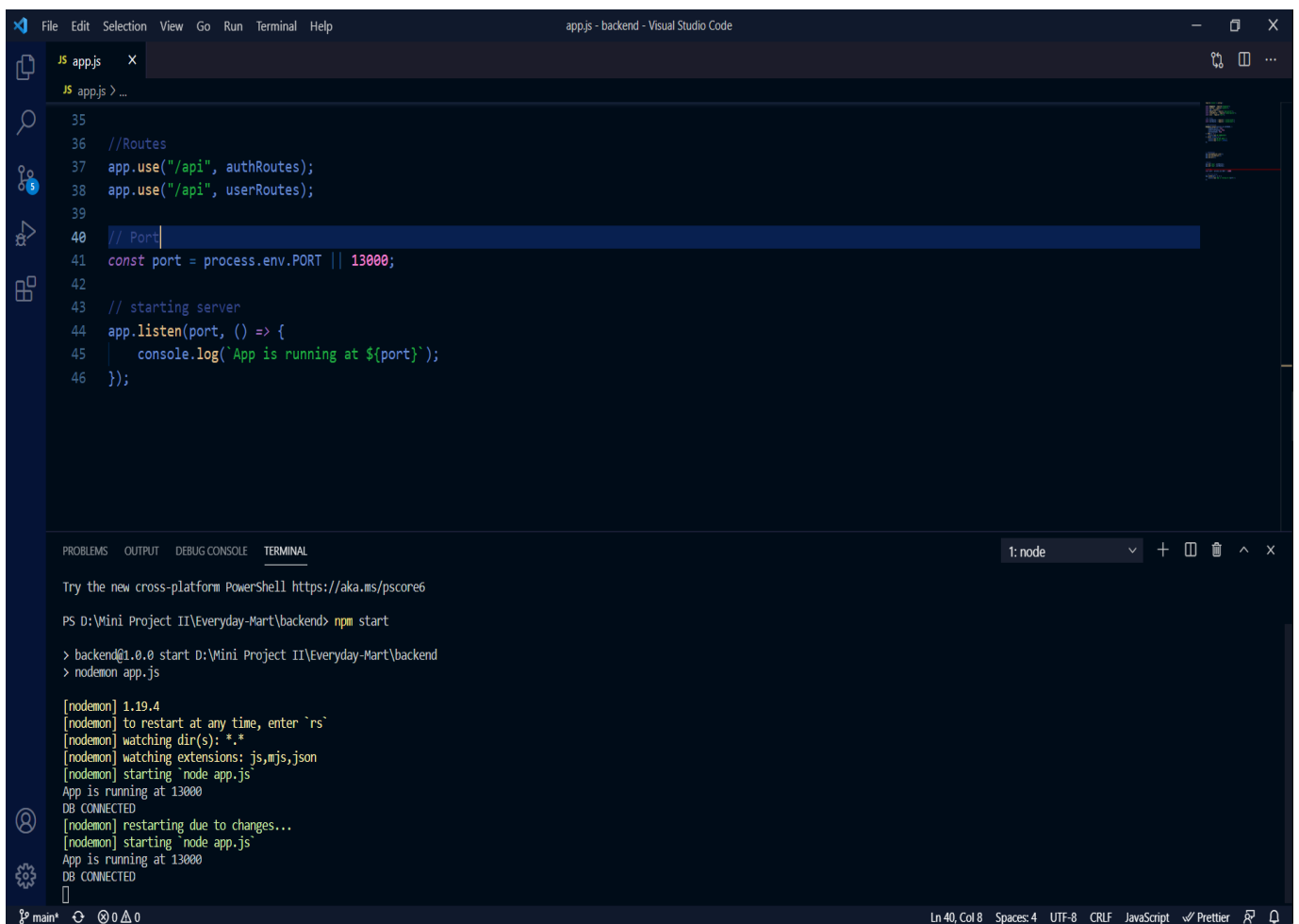
Email

Password

## Screenshots of Web Page Based Portal

### Backend Screenshot:

Successfully linked Project with MongoDB: -



```
File Edit Selection View Go Run Terminal Help
app.js - backend - Visual Studio Code

JS app.js x
JS app.js > ...

35
36 //Routes
37 app.use("/api", authRoutes);
38 app.use("/api", userRoutes);
39
40 // Port
41 const port = process.env.PORT || 13000;
42
43 // starting server
44 app.listen(port, () => {
45   console.log(`App is running at ${port}`);
46 });

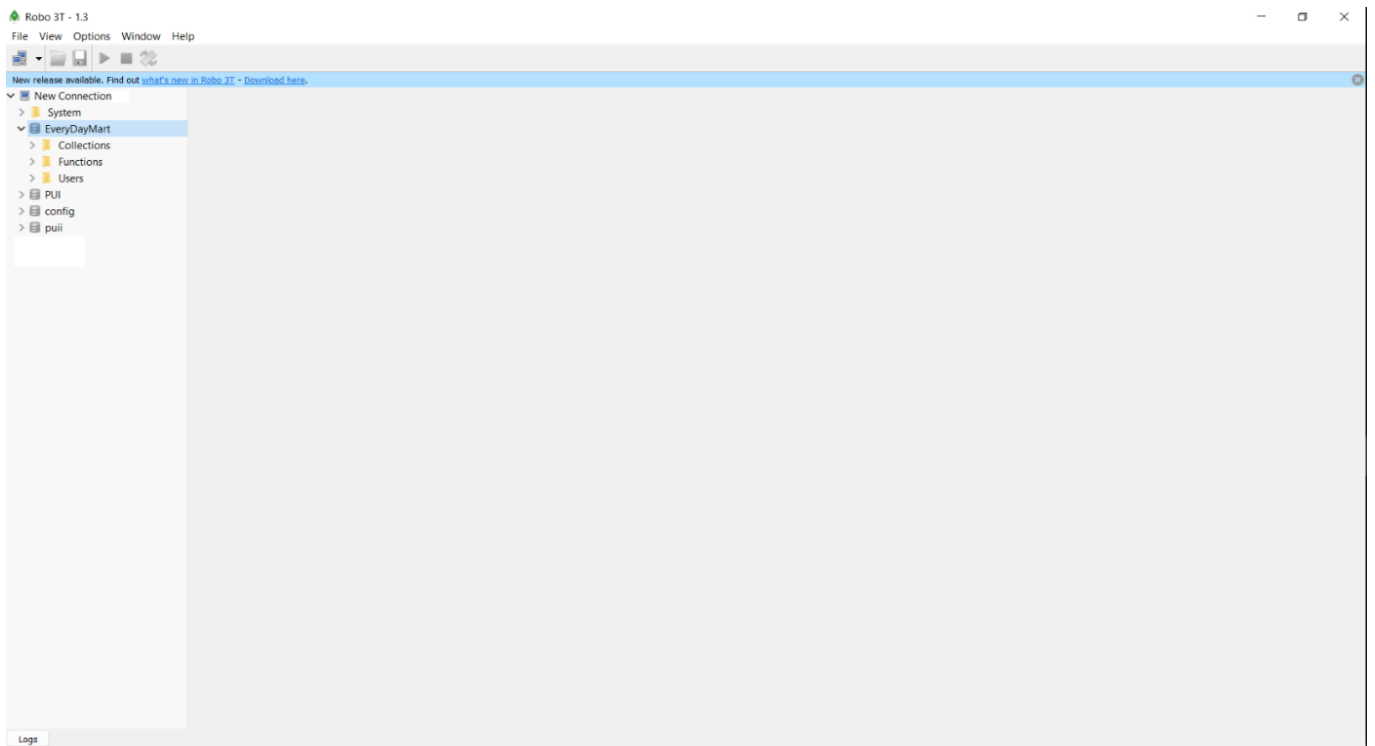
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: node

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Mini Project II\Everyday-Mart\backend> npm start

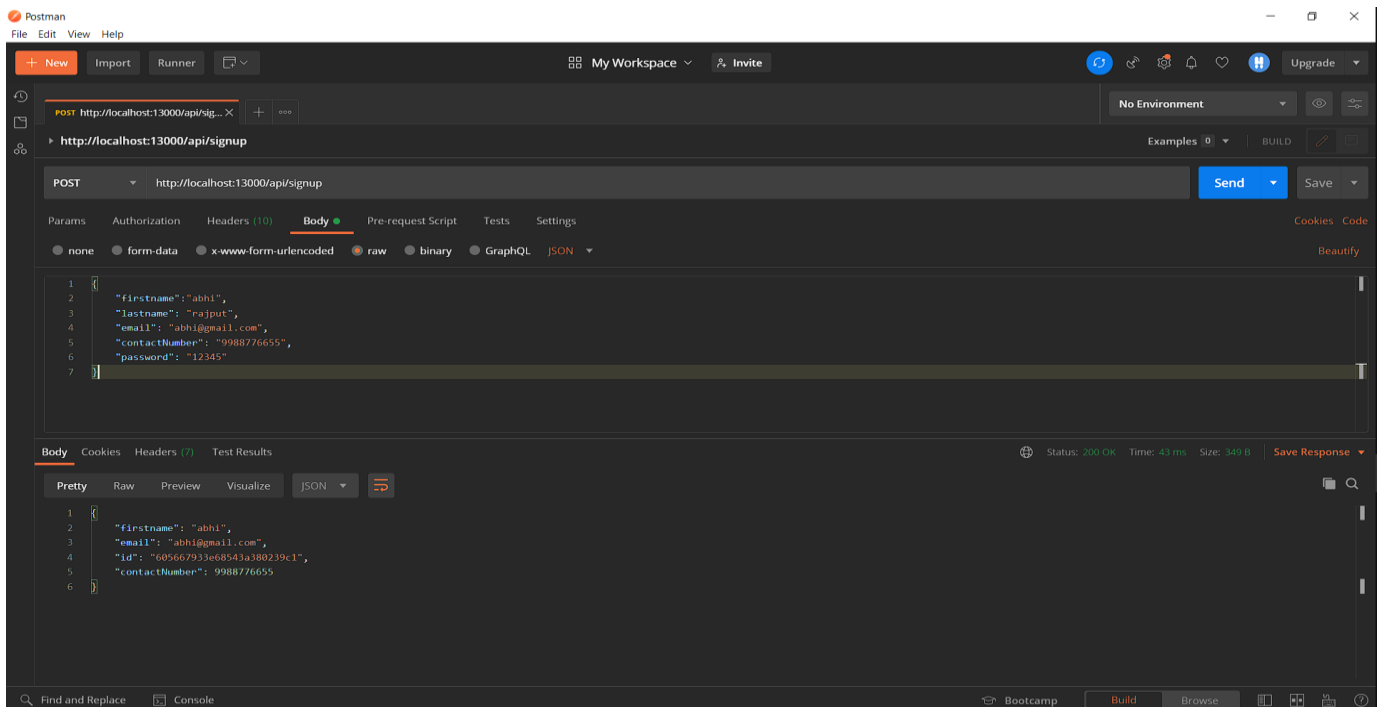
> backend@1.0.0 start D:\Mini Project II\Everyday-Mart\backend
> nodemon app.js

[nodemon] 1.19.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
App is running at 13000
DB CONNECTED
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
App is running at 13000
DB CONNECTED
[]
```

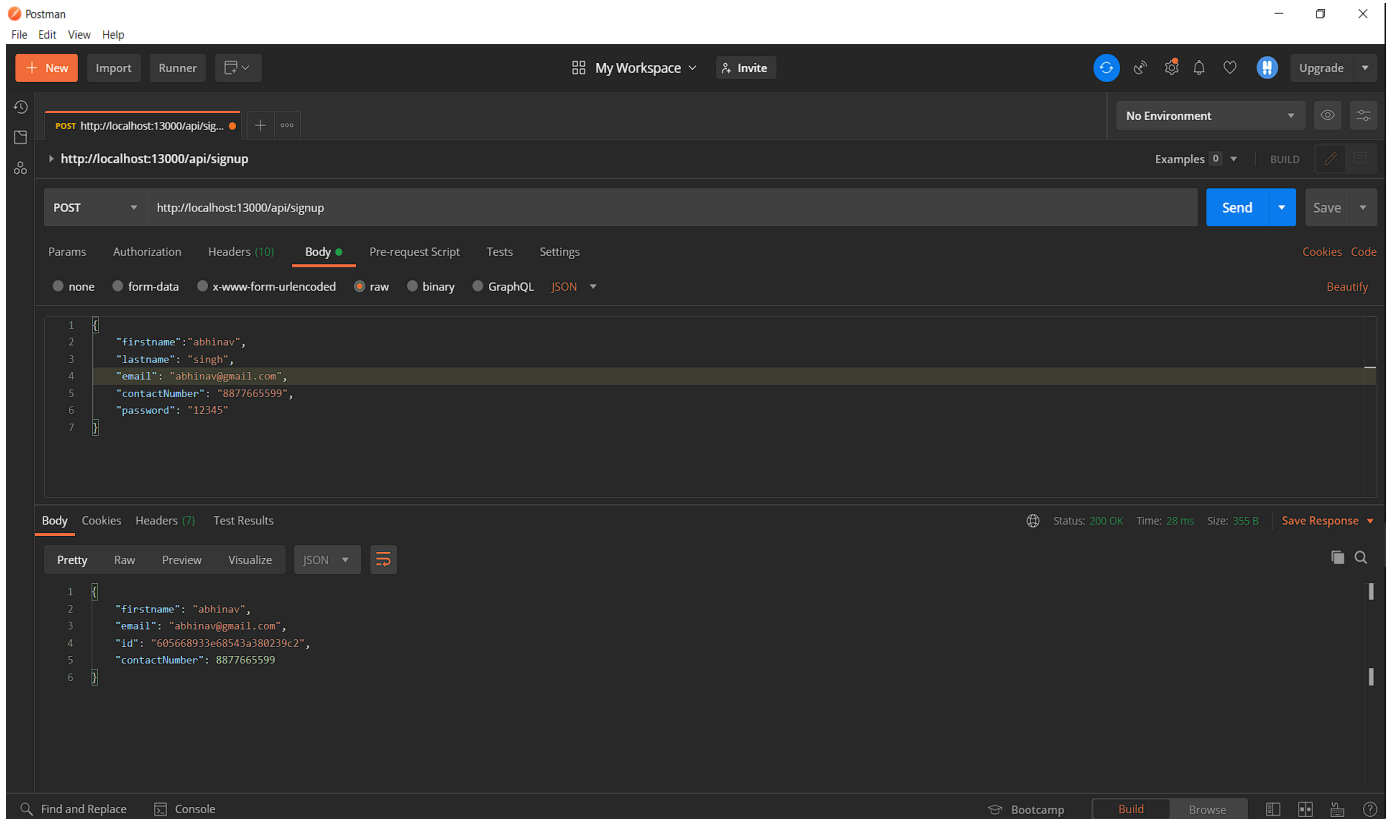


## Successful Signup using Postman: -

1<sup>st</sup> User



## 2<sup>nd</sup> User



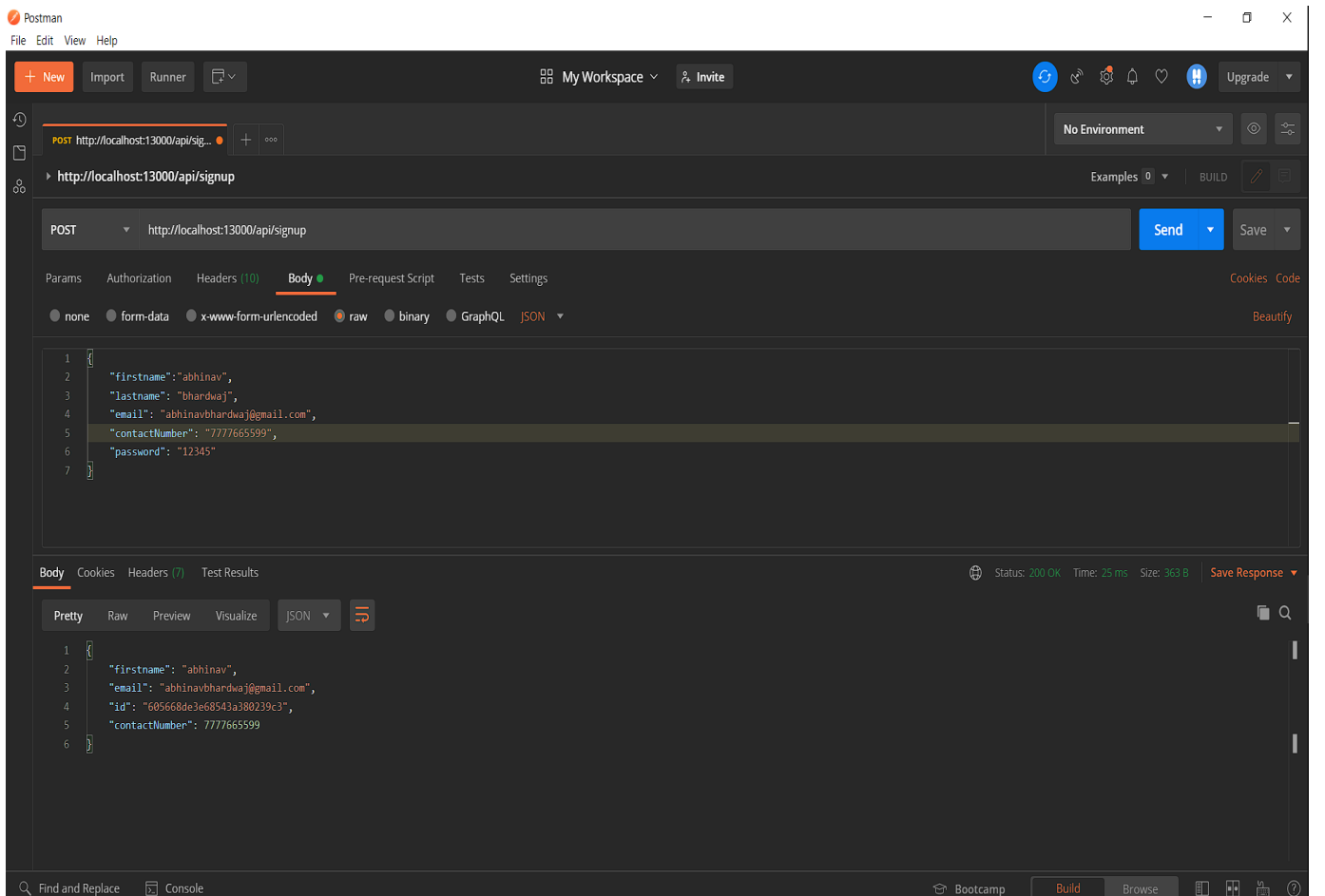
A screenshot of the Postman application showing a POST request to `http://localhost:13000/api/signup`. The request body is a JSON object with the following fields: `firstname` (abhinav), `lastname` (singh), `email` (abhinav@gmail.com), `contactNumber` (8877665599), and `password` (12345). The response is a JSON object with `firstname` (abhinav), `email` (abhinav@gmail.com), `id` (605668933e68543a380239c2), and `contactNumber` (8877665599). The status is 200 OK, time is 28 ms, and size is 355 B.

```
POST http://localhost:13000/api/signup
```

```
{
  "firstname": "abhinav",
  "lastname": "singh",
  "email": "abhinav@gmail.com",
  "contactNumber": "8877665599",
  "password": "12345"
}
```

```
{
  "firstname": "abhinav",
  "email": "abhinav@gmail.com",
  "id": "605668933e68543a380239c2",
  "contactNumber": "8877665599"
}
```

## 3<sup>rd</sup> User



A screenshot of the Postman application showing a POST request to `http://localhost:13000/api/signup`. The request body is a JSON object with the following fields: `firstname` (abhinav), `lastname` (bhardwaj), `email` (abhinavbhardwaj@gmail.com), `contactNumber` (7777665599), and `password` (12345). The response is a JSON object with `firstname` (abhinav), `email` (abhinavbhardwaj@gmail.com), `id` (605668de3e68543a380239c3), and `contactNumber` (7777665599). The status is 200 OK, time is 25 ms, and size is 363 B.

```
POST http://localhost:13000/api/signup
```

```
{
  "firstname": "abhinav",
  "lastname": "bhardwaj",
  "email": "abhinavbhardwaj@gmail.com",
  "contactNumber": "7777665599",
  "password": "12345"
}
```

```
{
  "firstname": "abhinav",
  "email": "abhinavbhardwaj@gmail.com",
  "id": "605668de3e68543a380239c3",
  "contactNumber": "7777665599"
}
```

Robo 3T - 1.3

File View Options Window Help

New release available. Find out [what's new in Robo 3T](#) - Download [here](#).

> New Connection db.getCollection('users').find().x

New Connection localhost:27017 EveryDayMart

db.getCollection('users').find({})

users 0.001 sec.

Key	Value	Type
(1) ObjectId("605667933e68543a380239c1")	{ 12 fields }	Object
_id	ObjectId("605667933e68543a380239c1")	ObjectId
role	0	Int32
purchases	[ 0 elements ]	Array
firstname	abhi	String
lastname	rajput	String
email	abhi@gmail.com	String
contactNumber	9988776655.0	Double
salt	5f3dd2c0-89c2-11eb-a5b4-47b977668456	String
encry_password	589e53d9551f627c97793a63036c61798bde54876ba49dfc2ec48cc0e435caa	String
createdAt	2021-03-20 21:22:27.955Z	Date
updatedAt	2021-03-20 21:22:27.955Z	Date
_v	0	Int32
(2) ObjectId("605668933e68543a380239c2")	{ 12 fields }	Object
_id	ObjectId("605668933e68543a380239c2")	ObjectId
role	0	Int32
purchases	[ 0 elements ]	Array
firstname	abhinav	String
lastname	singh	String
email	abhinav@gmail.com	String
contactNumber	8877665599.0	Double
salt	f75f70e0-89c2-11eb-a5b4-47b977668456	String
encry_password	0bf518dde261b3bd40b878199dac6f3ba8c21c5963998687100ca69a5d0a5b8c	String
createdAt	2021-03-20 21:26:43.183Z	Date
updatedAt	2021-03-20 21:26:43.183Z	Date
_v	0	Int32
(3) ObjectId("605668de3e68543a380239c3")	{ 12 fields }	Object
_id	ObjectId("605668de3e68543a380239c3")	ObjectId
role	0	Int32
purchases	[ 0 elements ]	Array
firstname	abhinav	String
lastname	bhardwaj	String
email	abhinavbhardwaj@gmail.com	String

Logs

Robo 3T - 1.3

File View Options Window Help

New release available. Find out [what's new in Robo 3T](#) - Download [here](#).

> New Connection db.getCollection('users').find().x

New Connection localhost:27017 EveryDayMart

db.getCollection('users').find({})

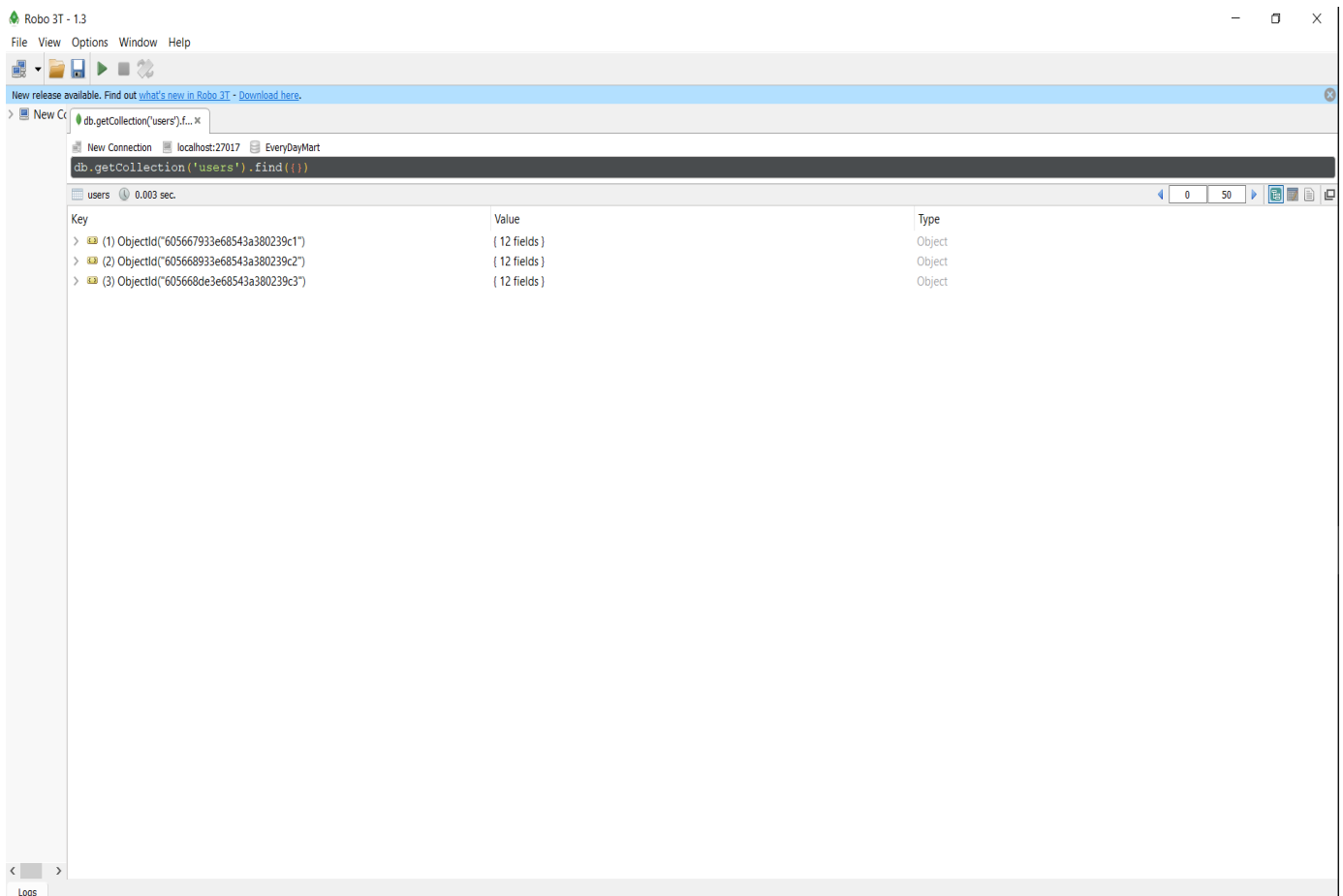
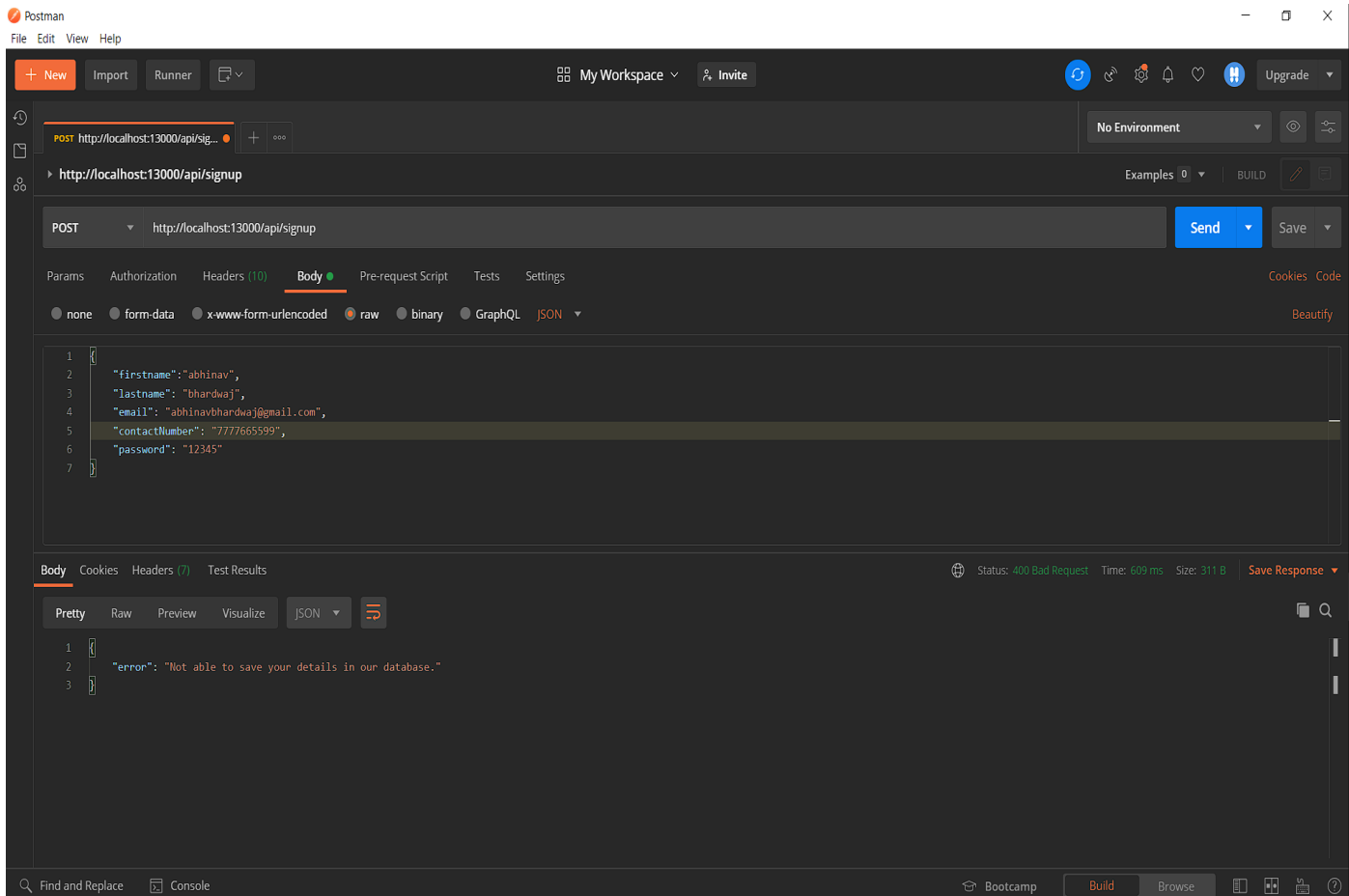
users 0.001 sec.

Key	Value	Type
contactNumber	9988776655.0	Double
salt	5f3dd2c0-89c2-11eb-a5b4-47b977668456	String
encry_password	589e53d9551f627c97793a63036c61798bde54876ba49dfc2ec48cc0e435caa	String
createdAt	2021-03-20 21:22:27.955Z	Date
updatedAt	2021-03-20 21:22:27.955Z	Date
_v	0	Int32
(2) ObjectId("605668933e68543a380239c2")	{ 12 fields }	Object
_id	ObjectId("605668933e68543a380239c2")	ObjectId
role	0	Int32
purchases	[ 0 elements ]	Array
firstname	abhinav	String
lastname	singh	String
email	abhinav@gmail.com	String
contactNumber	8877665599.0	Double
salt	f75f70e0-89c2-11eb-a5b4-47b977668456	String
encry_password	0bf518dde261b3bd40b878199dac6f3ba8c21c5963998687100ca69a5d0a5b8c	String
createdAt	2021-03-20 21:26:43.183Z	Date
updatedAt	2021-03-20 21:26:43.183Z	Date
_v	0	Int32
(3) ObjectId("605668de3e68543a380239c3")	{ 12 fields }	Object
_id	ObjectId("605668de3e68543a380239c3")	ObjectId
role	0	Int32
purchases	[ 0 elements ]	Array
firstname	abhinav	String
lastname	bhardwaj	String
email	abhinavbhardwaj@gmail.com	String
contactNumber	7777665599.0	Double
salt	24150f00-89c3-11eb-a5b4-47b977668456	String
encry_password	a902f887a34d90c4098cb0309fe2fadbcd92b019acaa2a7e7b34b820fc502da9	String
createdAt	2021-03-20 21:27:58.193Z	Date
updatedAt	2021-03-20 21:27:58.193Z	Date
_v	0	Int32

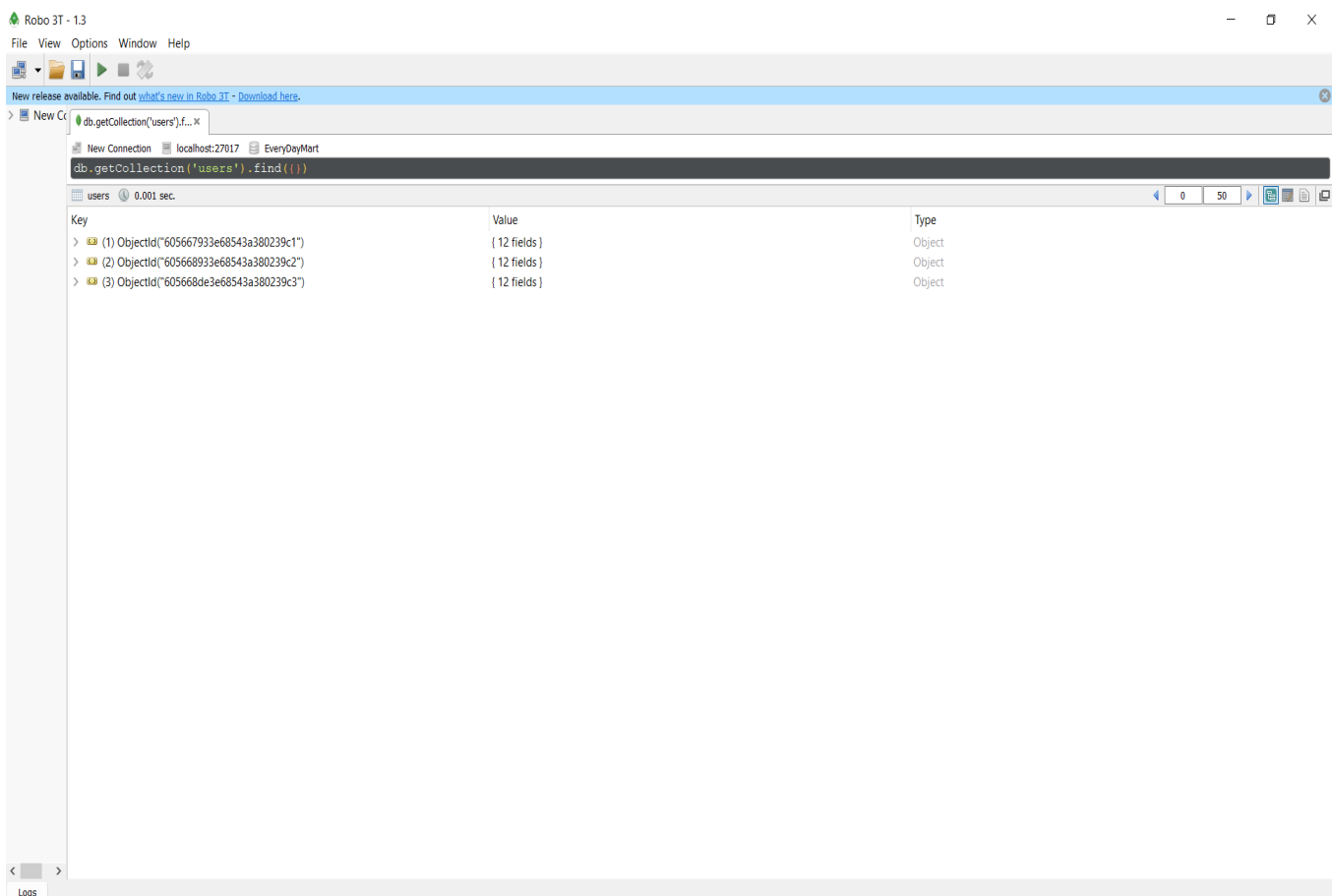
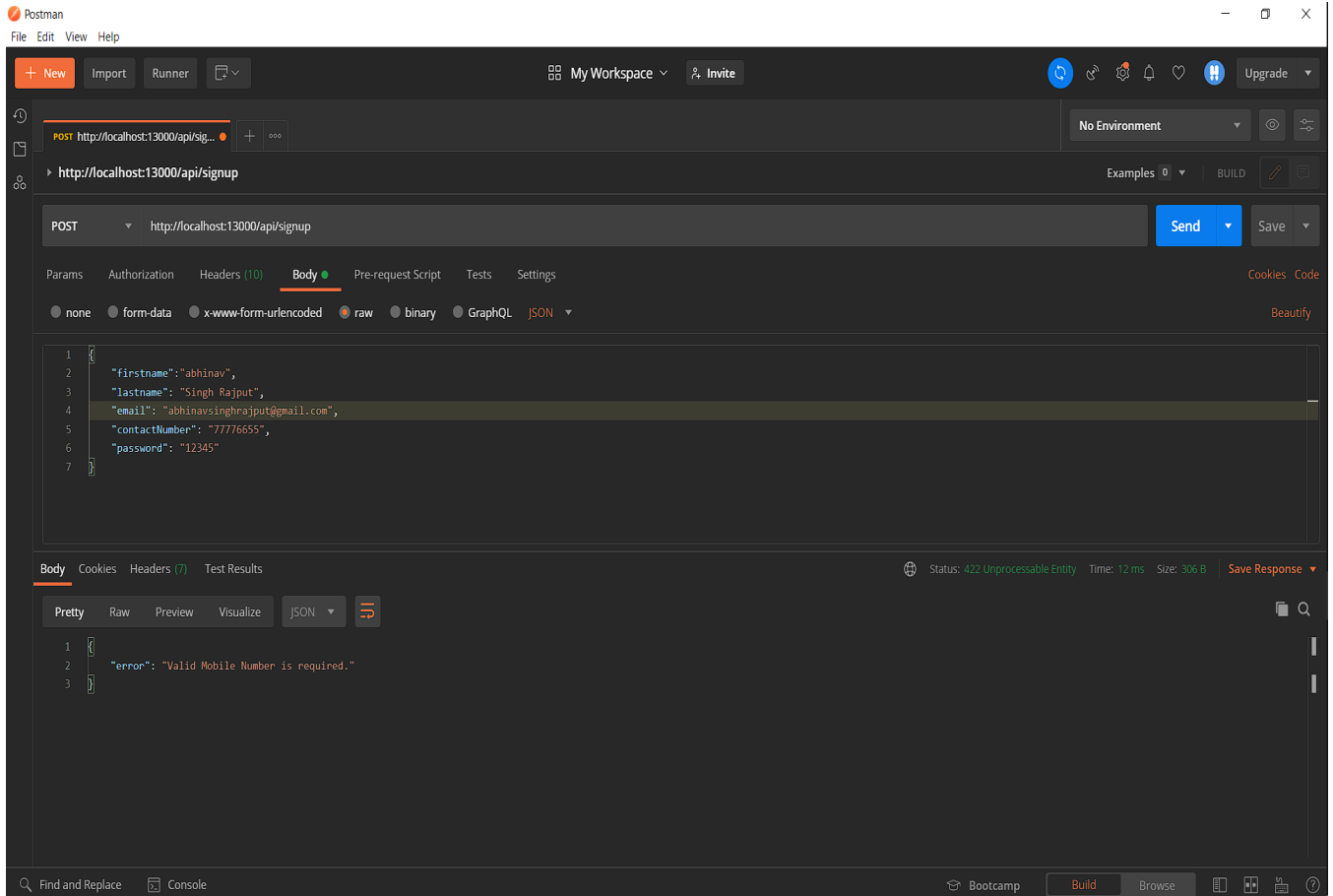
Logs



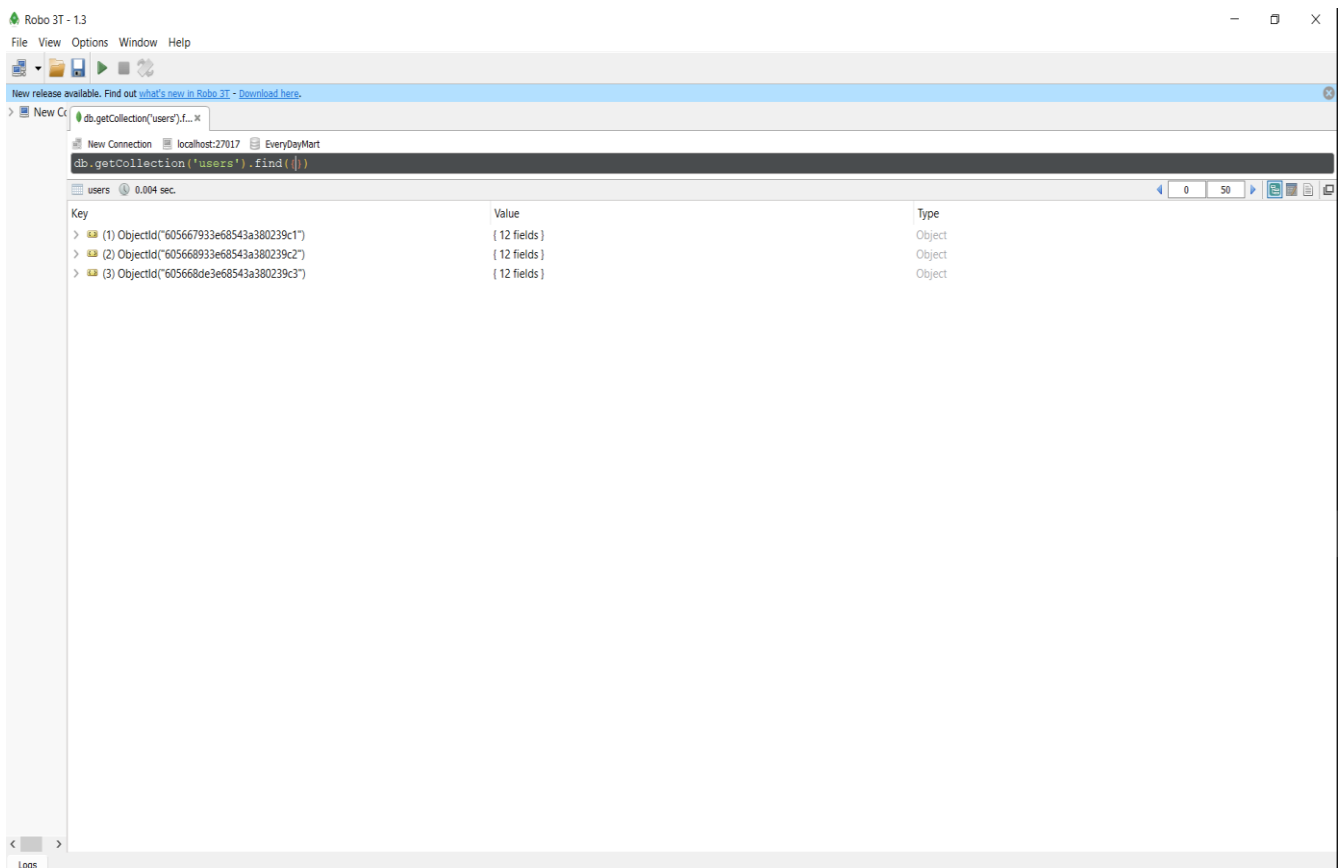
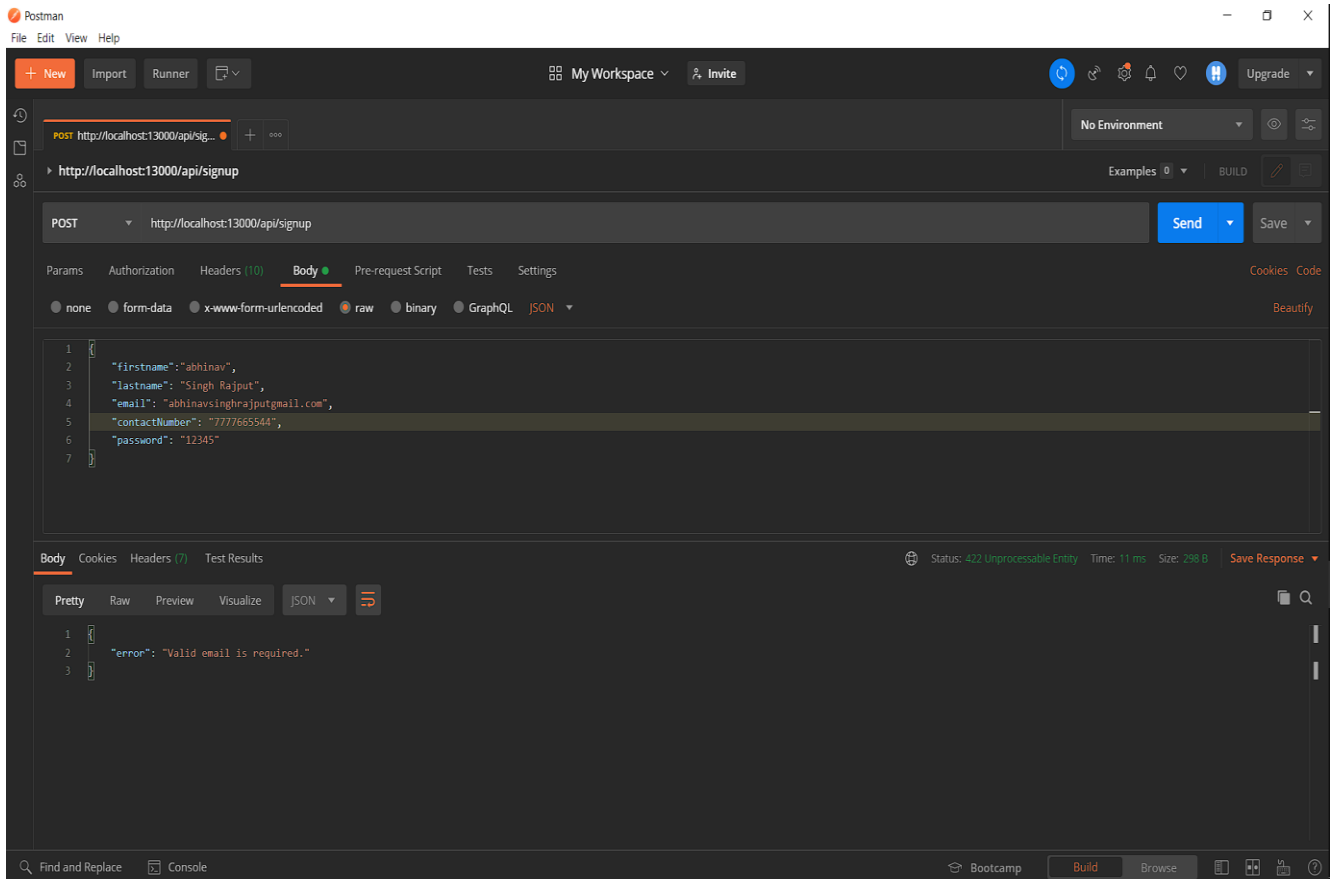
## If tried to sign up with duplicate data that is already in database, no entry in database



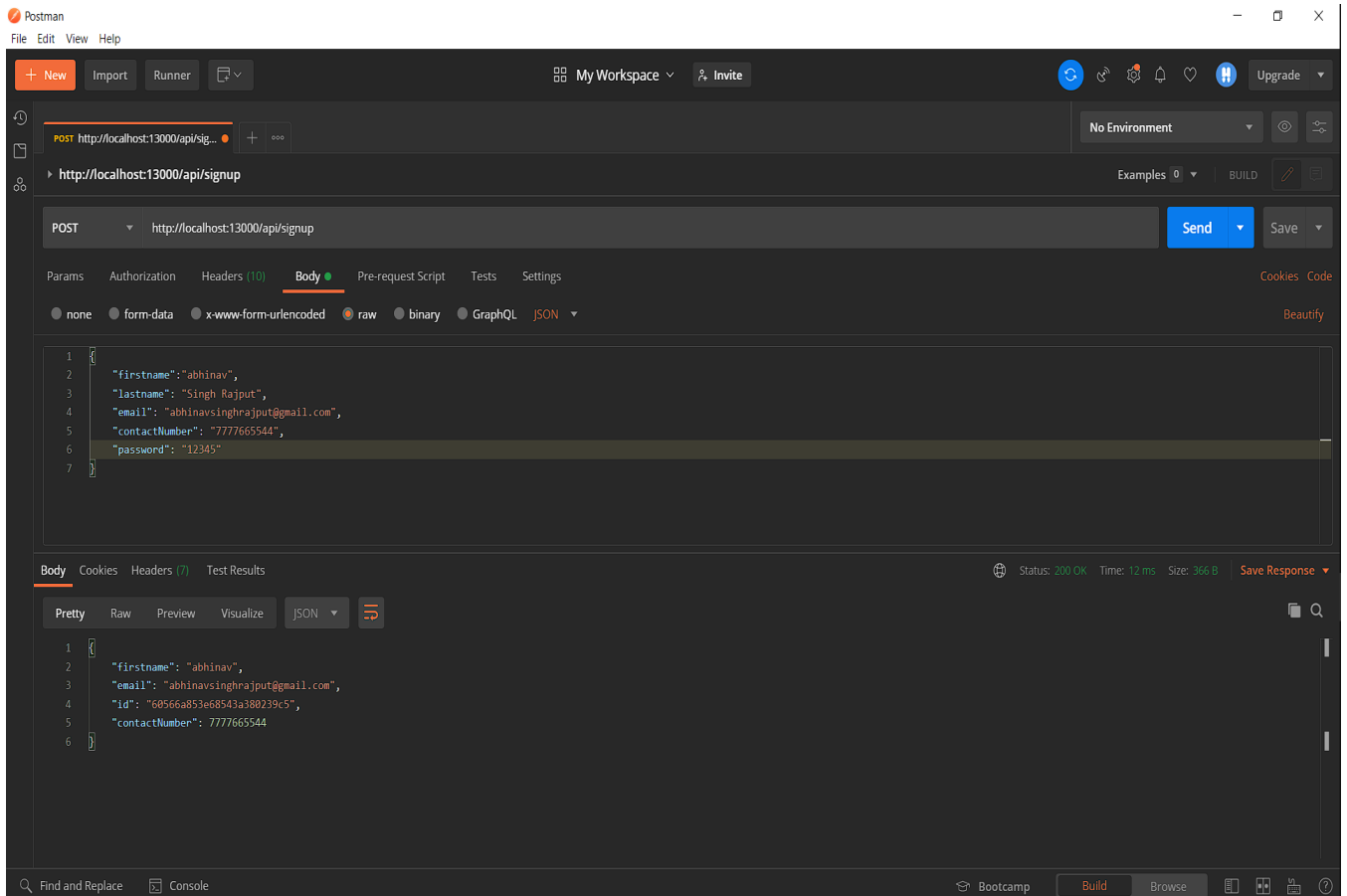
## If tried to sign up with invalid mobile number, no entry in database



## If tried to sign up with invalid email address, no entry in database



## And, again signup with proper details

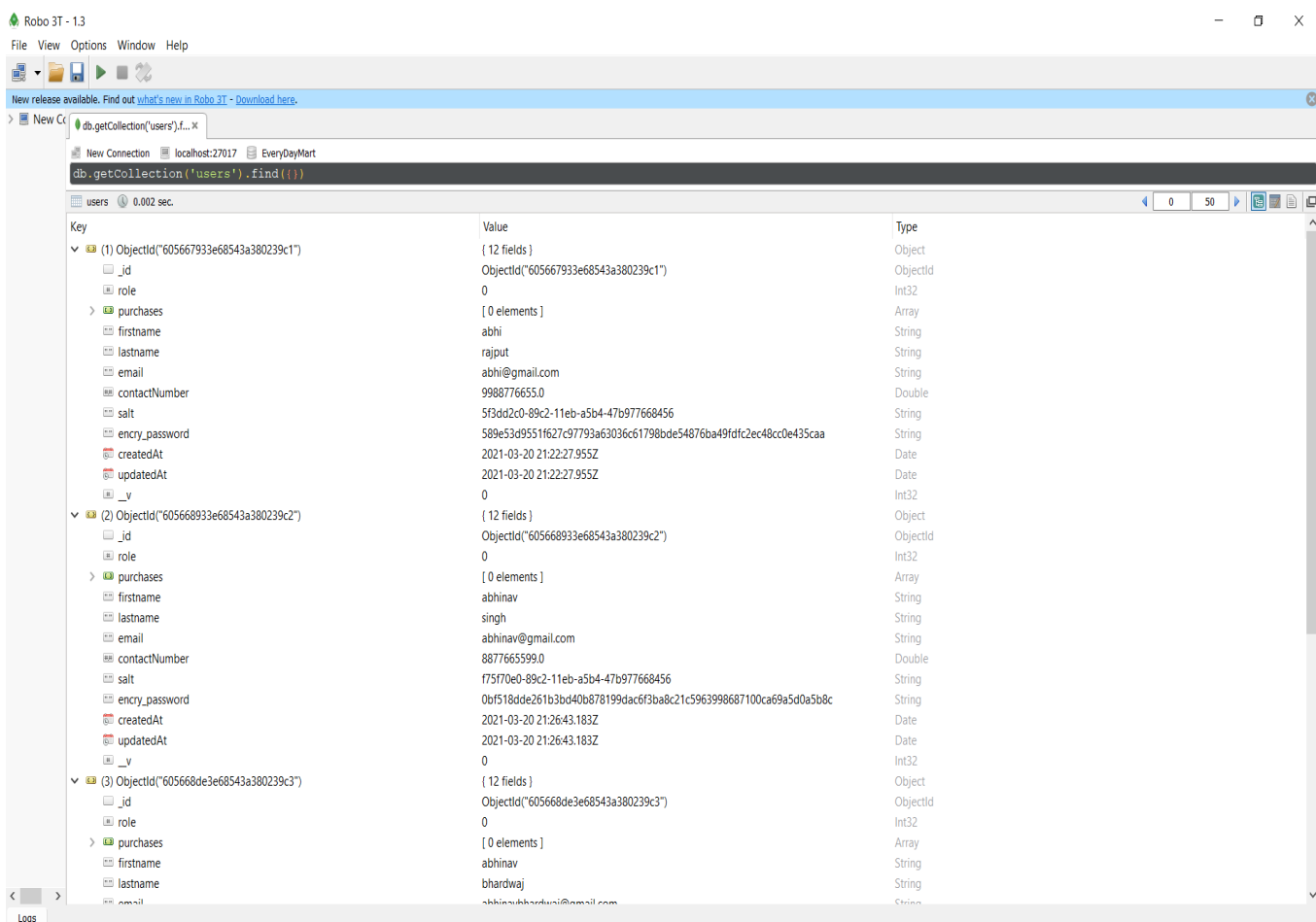


Postman interface showing a successful POST request to `http://localhost:13000/api/signup`. The request body is a JSON object with the following details:

```
{  "firstname": "abhinav",  "lastname": "Singh Rajput",  "email": "abhinavsinghrajput@gmail.com",  "contactNumber": "7777665544",  "password": "12345"}
```

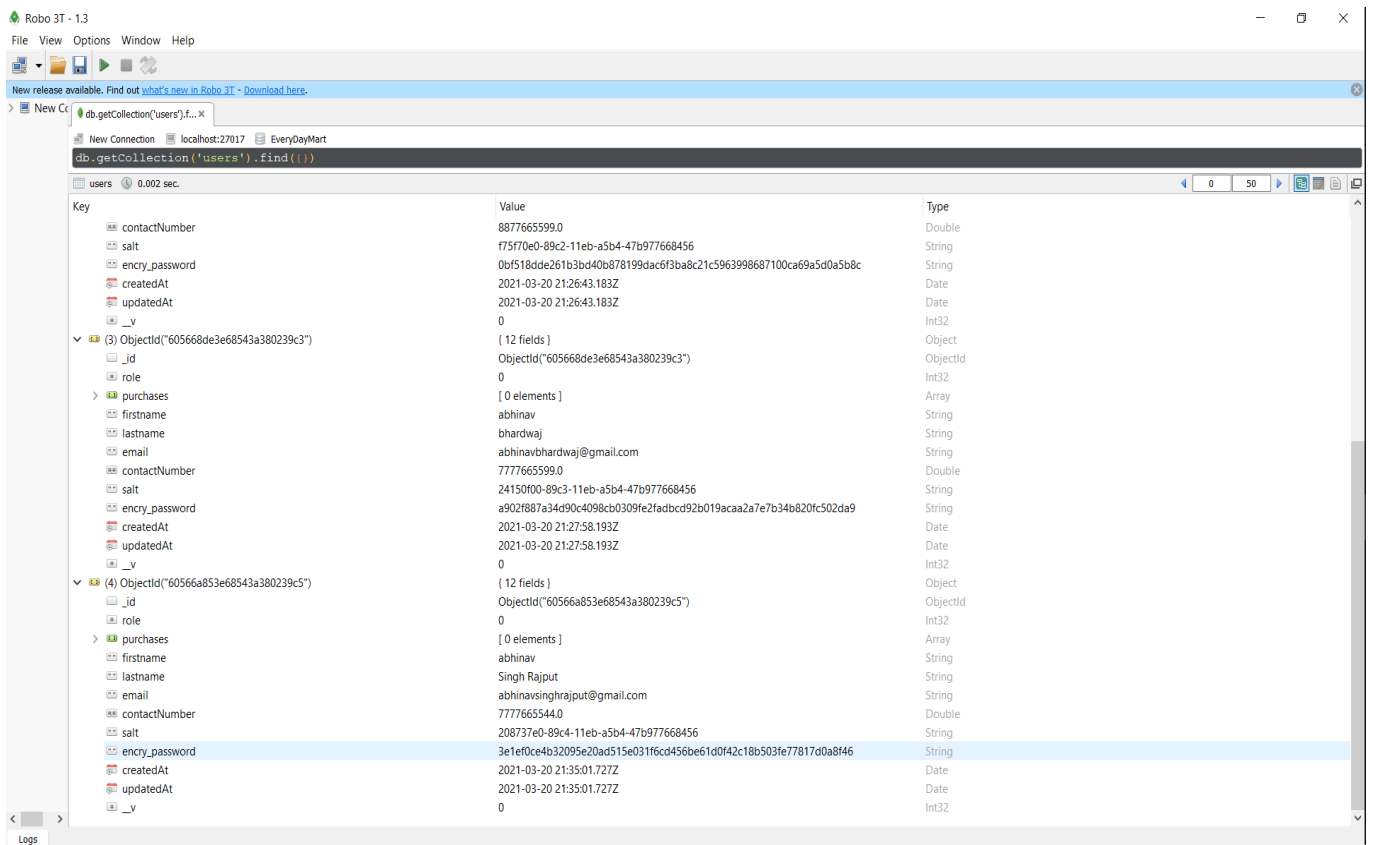
The response body is a JSON object with the following details:

```
{  "firstname": "abhinav",  "email": "abhinavsinghrajput@gmail.com",  "id": "60566a853e68543a380239c5",  "contactNumber": "7777665544"}
```

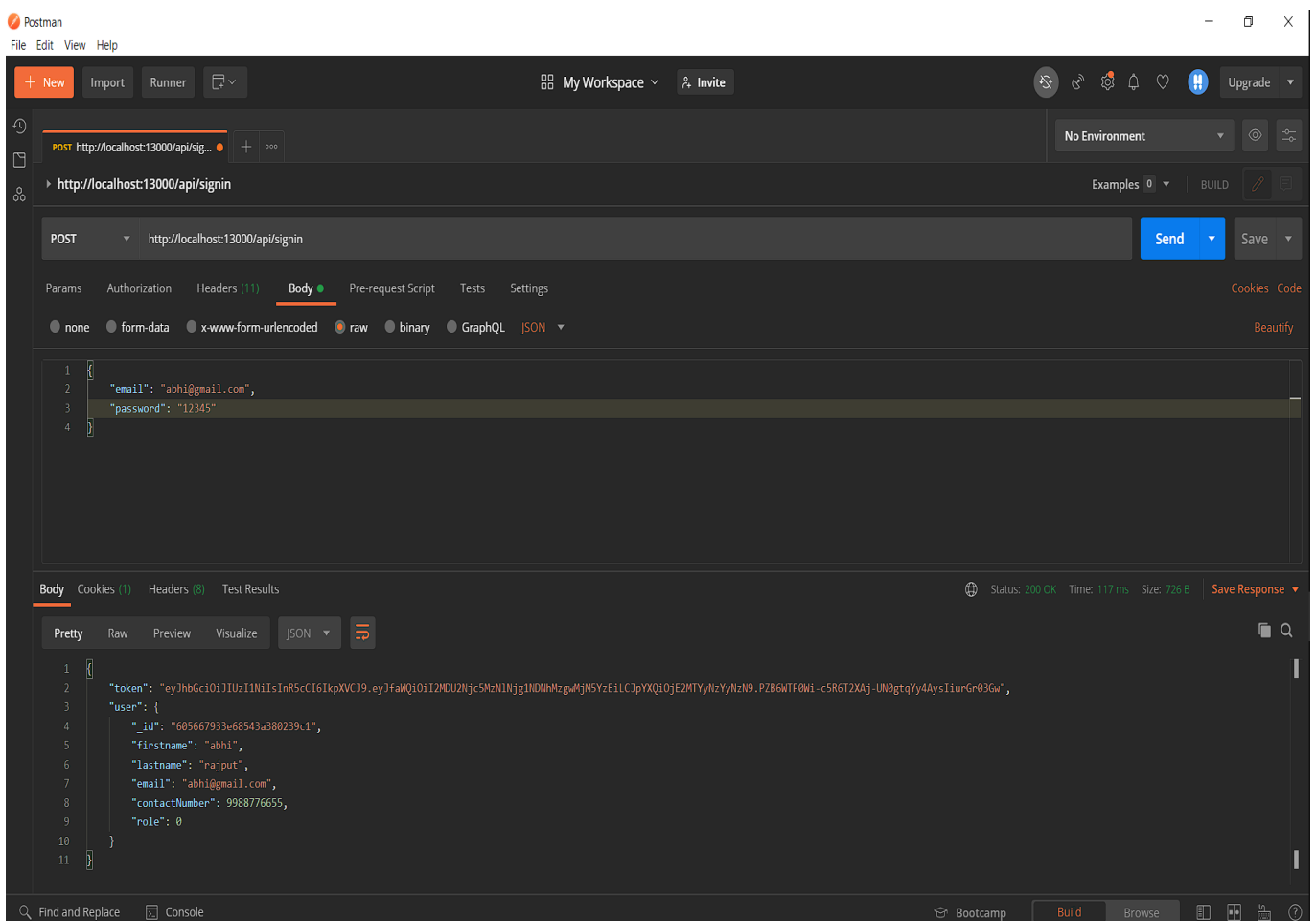


Robo 3T interface showing a successful query to the `users` collection. The query is `db.getCollection('users').find()`. The results show three user documents:

Key	Value	Type
(1) ObjectId("605667933e68543a380239c1")	{ 12 fields }	Object
id	ObjectId("605667933e68543a380239c1")	ObjectId
role	0	Int32
purchases	[ 0 elements ]	Array
firstname	abhi	String
lastname	rajput	String
email	abhi@gmail.com	String
contactNumber	9988776655.0	Double
salt	5f3dd2c0-89c2-11eb-a5b4-47b977668456	String
encry_password	589e53d9551f627c97793a63036c61798bde54876ba49dfc2ec48cc0e435caa	String
createdAt	2021-03-20 21:22:27.955Z	Date
updatedAt	2021-03-20 21:22:27.955Z	Date
_v	0	Int32
(2) ObjectId("605668933e68543a380239c2")	{ 12 fields }	Object
id	ObjectId("605668933e68543a380239c2")	ObjectId
role	0	Int32
purchases	[ 0 elements ]	Array
firstname	abhinav	String
lastname	singh	String
email	abhinav@gmail.com	String
contactNumber	8877665599.0	Double
salt	f75f70e0-89c2-11eb-a5b4-47b977668456	String
encry_password	0bf518dde261b3bd40b878199dac6f3ba8c21c5963998687100ca69a5d0a5b8c	String
createdAt	2021-03-20 21:26:43.183Z	Date
updatedAt	2021-03-20 21:26:43.183Z	Date
_v	0	Int32
(3) ObjectId("605668de3e68543a380239c3")	{ 12 fields }	Object
id	ObjectId("605668de3e68543a380239c3")	ObjectId
role	0	Int32
purchases	[ 0 elements ]	Array
firstname	abhinav	String
lastname	bhardwaj	String
email	abhinavbhardwaj@gmail.com	String



## Successful signed in with correct credentials



## Signing out

The screenshot shows the Postman interface with a GET request to `http://localhost:13000/api/signout`. The request is successful, returning a 200 OK status. The response body is displayed in JSON format:

```
{  "message": "User signout successfully"}
```

## Trying to sign in with incorrect email address that does not exist in database

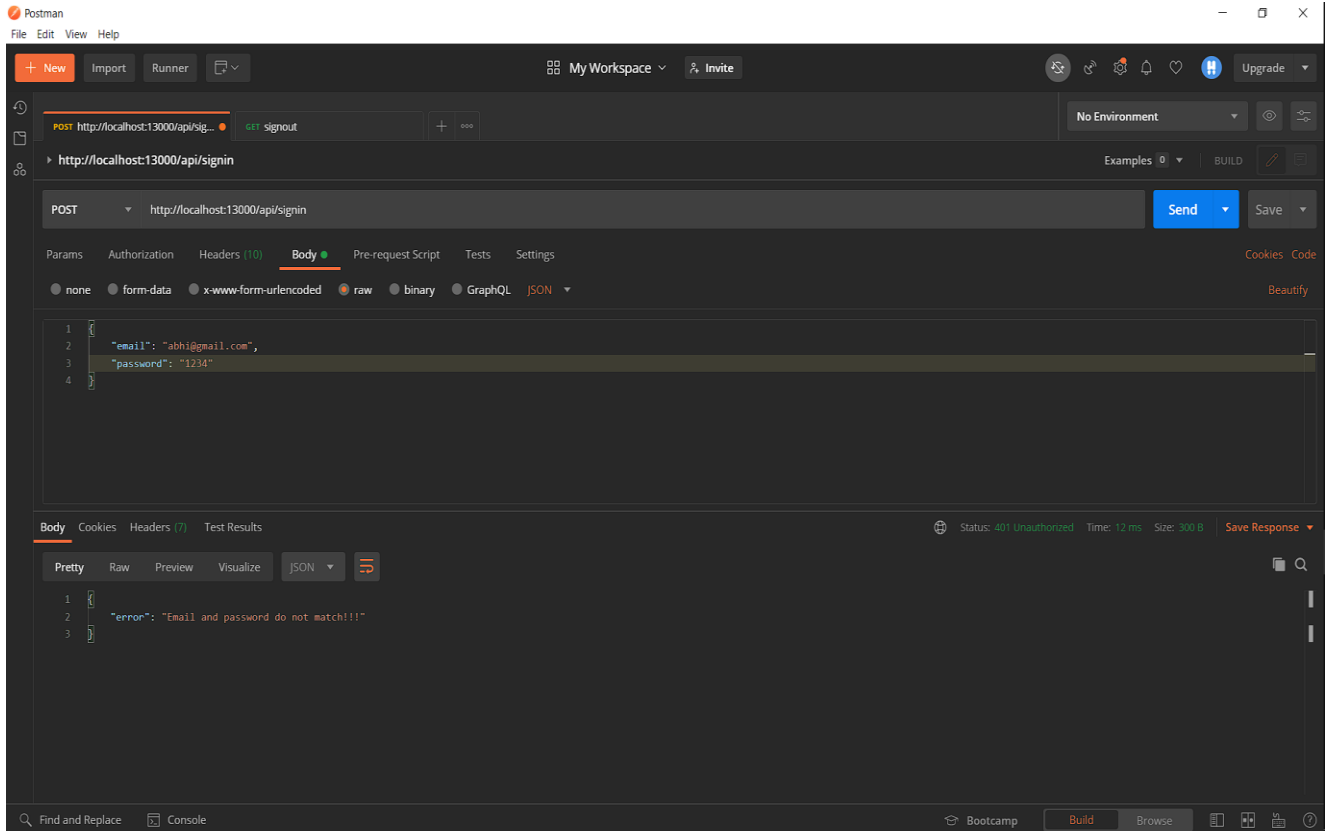
The screenshot shows the Postman interface with a POST request to `http://localhost:13000/api/signin`. The request body is:

```
{  "email": "abhiliii@gmail.com",  "password": "12345"}
```

The request fails with a 400 Bad Request status. The response body is displayed in JSON format:

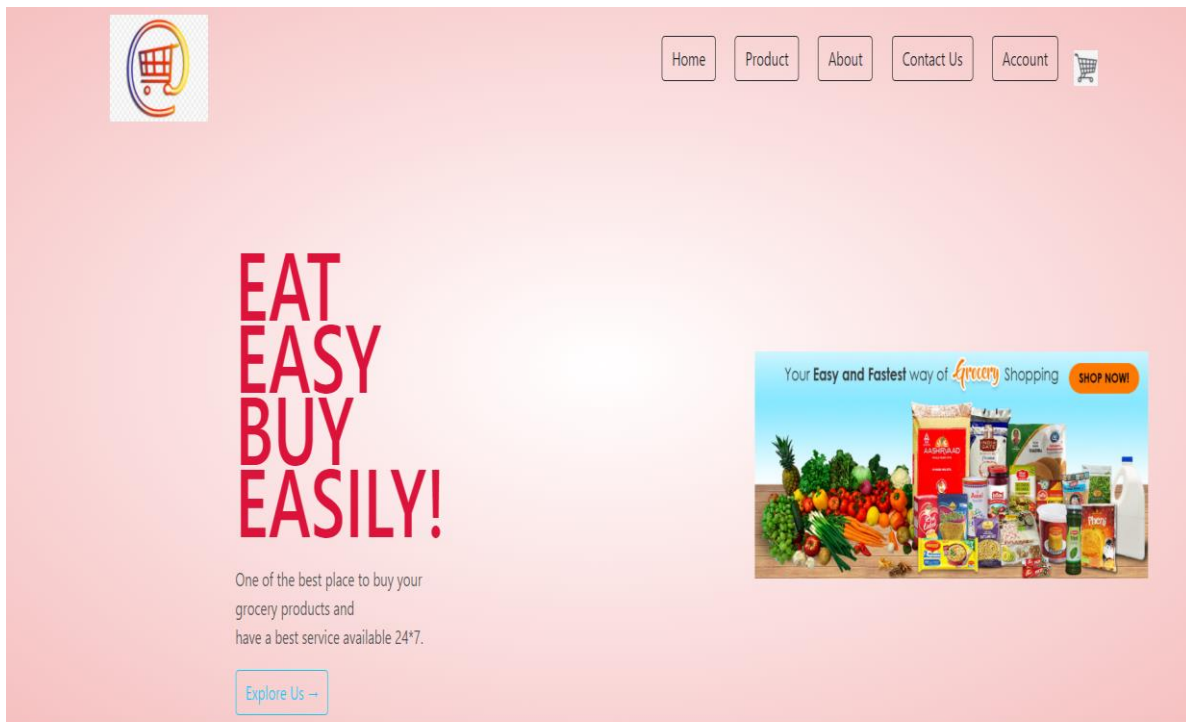
```
{  "error": "User's email does not exists!!!"}
```

# Trying to sign in with incorrect password



## Frontend Screenshots:

### Home Page:



### Categories section:



## Categories:-



Beverages



Grocery



Dairy Product



Others

## Products

### Product Section:

## Products:-



Maggi

★★★★☆  
₹ 50



Kurkure

★★★★☆  
₹ 20



Pepsi

★★★★★  
₹ 90



Bour-Bon

★★★★☆  
₹ 30

## New Products

New Product Section:

New Products:-



Park Avenue Perfume  
★★★★☆  
₹ 289



Veeba Pasta Sauce  
★★★★☆  
₹ 80



Dairy Milk Mouse  
★★★★☆  
₹ 150



Bour-Bon  
★★★★☆  
₹ 50

Exclusive Available on EveryDay Mart

Tea

Offer Section:

Exclusive Available on EveryDay Mart




UP TO 30% OFF  
Tea & Coffee store


Tea  
And  
Coffee  
Products


Good healthy products for Daily Morning  
Routine and It will keep you fresh all the  
day. Different Company Tea and Coffee are  
available at 30% discount.


Buy Now →

Footer Section:









### Download Our App

Download App for Android and ios mobile phone.



Our purpose is to sell our store product and give a good and convinient service to our customer.


### Useful Links

- Coupons
- Blog Post
- Return Policy
- Join Affiliate

### Follow Us

- Facebook
- Twitter
- Instagram
- YouTube

All Product Section:




Home

Product

About

Contact Us

Account



### All Products

Default Sorting ▾



Maggi

★★★★☆

₹ 50



Kurkure

★★★★☆

₹ 20



Pepsi

★★★★★

₹ 90



Bour-Bon

★★★★☆

₹ 30



Park Avenue Perfume

★★★★☆  
₹ 289



Veeba Pasta Sauce

★★★★☆  
₹ 80



Dairy Milk Mouse

★★★★☆  
₹ 150



Bour-Bon

★★★★☆  
₹ 50

1 2 3 4 →

### Download Our App

Download App for Android and ios mobile



Our purpose is to sell our store product

### Useful Links

- Coupons
- Blog Post

### Follow Us

- Facebook
- Twitter

## Individual Product Detail:

# ParkAvenue Perfume



280 rs

1

Add To Cart

## Product detail

It is a very good perfume offered by  
the park avenue.

