Lambton College

CSD-3102 Full Stack JavaScript

Student Name:	Kisan Rai
Student Number:	C0910925
Assignment name:	MERN Final Assignment
Instructor name:	Prof. Terry D.Silva
Submission Date	03rd -Dec-2024

Table of Contents

Acknowledgements

I would like to express my sincere gratitude to my professor, Terry D. Silva, for his invaluable support and guidance throughout this project. I deeply appreciate the resources, materials, and extra information provided, which greatly contributed to my understanding and development of the project. Professor Silva's continuous encouragement and thoughtful feedback helped me navigate challenges and successfully complete this assignment. His expertise and willingness to assist me at every step were instrumental in my learning experience. Thank you for your dedication and support.

1. Introduction

The purpose of this project is to build a comprehensive full-stack web application using the **MERN stack**, which includes **MongoDB**, **Express.js**, **React.js**, and **Node.js**. This application is designed to manage user information, providing features for adding, viewing, updating, and deleting user data. As part of the Full Stack JavaScript course for Fall 2024, this project aims to showcase the integration of both front-end and back-end technologies to create a functional and user-friendly application.

The project serves as a practical demonstration of my ability to apply JavaScript in building a complete web application that adheres to real-world business needs. This project involves working with databases, building RESTful APIs, and creating a user interface that is interactive and responsive. In addition to technical implementation, the project also provides an opportunity to focus on best practices, code structure, and documentation.

2. Project Objectives

The primary goal of this project is to develop a **MERN** stack application that satisfies several key requirements:

- **User Data Management**: The application should allow users to add, view, update, and delete personal information through a web interface.
- **Front-end Development**: Utilize **React.js** to build a dynamic and responsive interface for interacting with the backend.
- **Back-end Development**: Use **Express.js** to handle HTTP requests and **MongoDB** to store user data.
- **Authentication**: Implement user login functionality to authenticate users before allowing access to sensitive data.
- **Data Display**: Integrate a data grid, using tools such as **ag-Grid**, to present user information in a tabular format.
- **Styling**: Use **Bootstrap/W3CSS** to ensure the application is visually appealing and responsive across different devices.

3. Technologies Used

3.1 Frontend:

- React.js: The core JavaScript library used to build the user interface of the web application. React's component-based architecture allows for the creation of reusable components, making it easier to manage the UI.
- Axios: A promise-based HTTP client used to send asynchronous requests from the React frontend to the Express backend. It was used to fetch and send data to the server for CRUD operations.
- ag-Grid: A powerful grid library that displays user data in a sortable, filterable, and interactive table format. It supports features like pagination and data sorting out of the box.
- Bootstrap/W3CSS: Used for styling the user interface. Bootstrap provides responsive grid layouts and components like forms, buttons, and navigation bars, while W3CSS was used for additional visual styling to ensure a clean and modern look.
- Evergreen UI Components: Optional third-party React components used to enhance the user experience, providing pre-designed elements for forms, buttons, and dialogs.

3.2. Backend:

- Node.js: A JavaScript runtime environment used to build the backend of the application. Node.js allows for asynchronous processing, making it ideal for handling concurrent requests in web applications.
- Express.js: A minimal and flexible web application framework for Node.js that provides a set of tools to build APIs. Express was used to handle the routing, manage HTTP requests, and interact with the MongoDB database.
- MongoDB: A NoSQL database used to store user data in a flexible JSON format. MongoDB was chosen because of its ability to scale and handle large volumes of unstructured data efficiently.
- Mongoose: An Object Data Modeling (ODM) library for MongoDB and Node.js. Mongoose was used to define the schema for the user data and interact with the database.

- Login.js: A library used to hash passwords securely, providing an extra layer of protection for sensitive user information.

4. Application Features

4.1. User Authentication (Login Page)

The login page serves as the entry point to the application. It requires users to authenticate themselves using a username {'**test@estexample.com'**} and password{'**password123**'}. Once authenticated, users can access the application's core features such as adding, updating, and viewing user data. This feature ensures that only authorized individuals can modify user information.

4.2. User Management (Add, Update, Delete)

The core functionality of the application revolves around user management. There are three key actions:

- **Add User**: Users can fill in a form with their personal details such as first name, last name, date of birth, email, phone number, address, and user notes. This data is sent to the backend where a new record is created in the MongoDB database.
- **Update User**: Users can select an existing record to view and update. The form is pre-populated with the current details, and changes can be saved by submitting the updated form.
- **Delete User**: Users can delete a user's data from the system. This action permanently removes the record from the database.

4.3. View Users (Data Grid)

The application displays a list of all users stored in the MongoDB database. This data is presented in a table format using **ag-Grid**. The grid allows users to:

- Sort and visualize records.

- View details of each user by selecting a record.
- Navigate between pages of data, making it easier to manage large amounts of user information.

4. Responsive UI

The application is fully responsive, meaning that it adapts to different screen sizes. This is achieved using **Bootstrap/W3CSS**, which ensures that the UI elements adjust accordingly whether the application is accessed on a desktop, tablet, or mobile device.

5. Backend Architecture

The backend of the application is responsible for handling HTTP requests from the frontend. It provides the necessary API endpoints for performing CRUD operations on user data. The backend architecture follows the **Model-View-Controller (MVC)** pattern, where:

- **Model**: Defines the structure of the user data in MongoDB.
- **View**: Serves as the client-side UI for interacting with the user data.
- **Controller**: Handles the logic for performing CRUD operations, such as adding, updating, and deleting users.

The server is set up using **Express.js**, and the database connection is established with **MongoDB** through **Mongoose**. The API endpoints interact with the database to retrieve, insert, update, or delete user records.

6. Populating MongoDB Database with Random Data

As part of the project requirements, the MongoDB database needed to be populated with at least 10 random records of user data. To ensure that the application would function with actual data, I utilized a method to automatically populate the database with placeholder data using a data generation tool.

Data Population Process:

For the population of random data, I used one of the following tools:

- **JSONPlaceholder**: A free API that provides fake data for testing and prototyping. This API was used to generate random user records that contained fields such as first name, last name, date of birth, address, phone number, email, and notes.
- **Faker.js**: This tool was used to generate random data directly in the backend to insert into MongoDB. Faker.js allows for the creation of realistic-looking names, addresses, dates, and other user-related information.
- **PopulateData.json**: To streamline the process, I created a **populateData.json** file in the backend directory. This file contained a set of random user data entries in JSON format. The file could be used to populate the MongoDB database when the application was initialized or when a manual trigger was required to populate the database with dummy data.

7. Challenges Faced

1. Data Synchronization Between Frontend and Backend

One of the challenges was ensuring that the data between the frontend and backend remained in sync, particularly when adding, updating, or deleting user records. This was resolved by using **Axios** to send requests to the backend and using **React's state management** (via useState and useEffect) to keep the frontend updated in real-time.

2. Managing Complex Forms in React

Another challenge was managing the state of complex forms in React. Each form (for adding or updating a user) contained multiple fields, and ensuring that the data entered was correctly captured, validated, and sent to the backend was a significant challenge. React's controlled components and state management via **useState** hooks helped in organizing and handling form data effectively.

3. Designing the User Interface

Creating a user-friendly interface that was both visually appealing and easy to navigate was an important aspect of the project. While building the UI, I focused on clear navigation, easy form submission, and responsiveness across devices. I used **Bootstrap** to help with styling and layout, but also customized it to fit the unique requirements of the project.

8. Conclusion

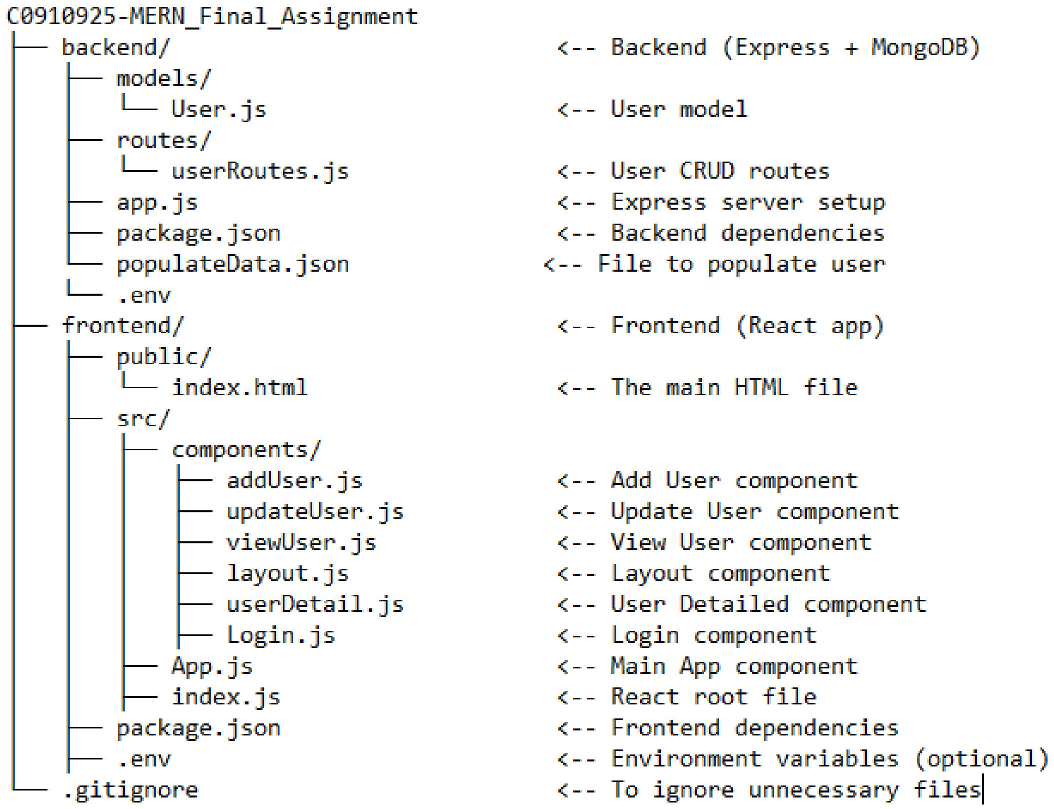
In conclusion, this full-stack JavaScript project successfully meets the requirements and demonstrates the integration of various technologies to create a functional web application. The application allows users to manage their data efficiently with features for adding, viewing, updating, and deleting user records. By using **React.js** for the frontend and **Express.js** with **MongoDB** for the backend, the project showcases the power of the **MERN stack** in building modern web applications.

This project helped me enhance my understanding of full-stack development, particularly in areas such as asynchronous programming, database management, and UI design. Despite facing challenges such as data synchronization, form management, and user authentication, I was able to find effective solutions using the tools and libraries available within the JavaScript ecosystem.

The application is fully functional, user-friendly, and easily extendable, making it a strong example of my capability to develop full-stack applications. This project not only fulfilled the assignment requirements but also provided valuable learning experiences that I can apply in future development projects.

9. Program Instructions:

Here is what my entire project directory looks like, with the project divided into two parts: **backend** and **frontend**.

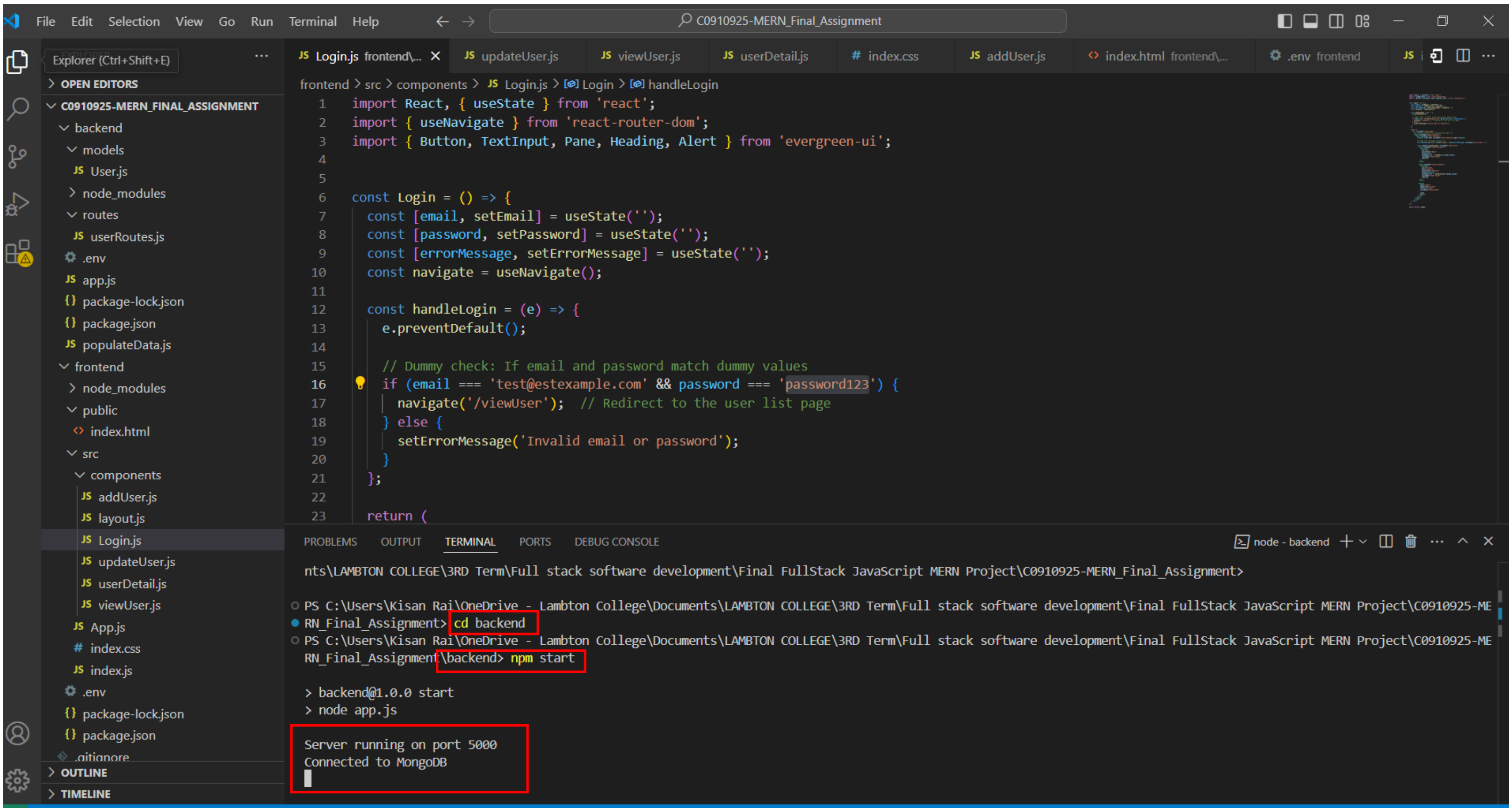


```
C0910925-MERN_Final_Assignment
├── backend/                            <-- Backend (Express + MongoDB)
│   ├── models/
│   │   └── User.js                     <-- User model
│   ├── routes/
│   │   └── userRoutes.js               <-- User CRUD routes
│   ├── app.js                          <-- Express server setup
│   ├── package.json                    <-- Backend dependencies
│   ├── populateData.json              <-- File to populate user
│   └── .env
├── frontend/                           <-- Frontend (React app)
│   ├── public/
│   │   └── index.html                  <-- The main HTML file
│   ├── src/
│   │   ├── components/
│   │   │   ├── addUser.js              <-- Add User component
│   │   │   ├── updateUser.js           <-- Update User component
│   │   │   ├── viewUser.js             <-- View User component
│   │   │   ├── layout.js               <-- Layout component
│   │   │   ├── userDetail.js           <-- User Detailed component
│   │   │   ├── Login.js                <-- Login component
│   │   ├── App.js                      <-- Main App component
│   │   ├── index.js                    <-- React root file
│   ├── package.json                    <-- Frontend dependencies
│   ├── .env                            <-- Environment variables (optional)
└── .gitignore                          <-- To ignore unnecessary files
```

10 To run the Project

10.1 Backend server

To run the backend server, first open the terminal and navigate to the backend directory by typing [**cd backend**] and hitting Enter. This will change the directory to the backend folder. Then, type [**npm start**] to run the backend server.

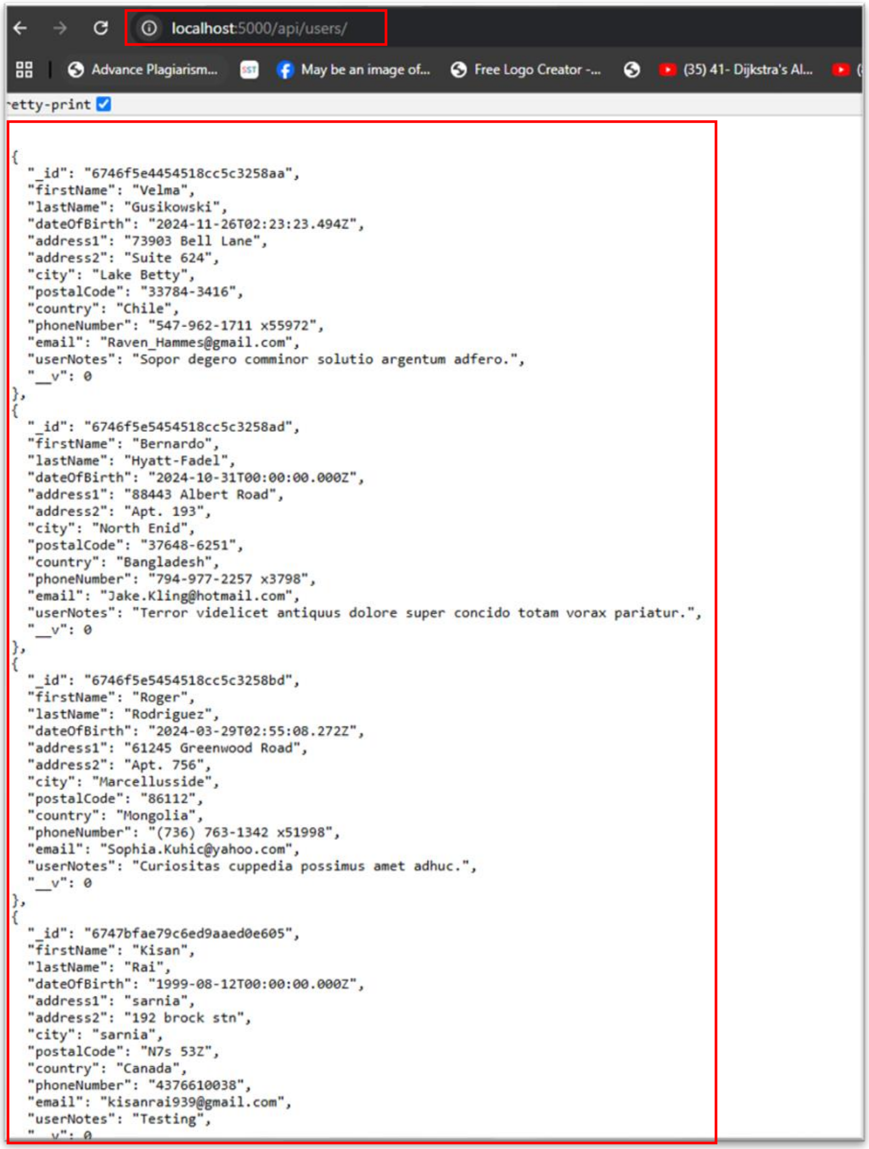


```
import React, { useState } from 'react';
import { useNavigate } from 'react-router-dom';
import { Button, TextInput, Pane, Heading, Alert } from 'evergreen-ui';


const Login = () => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [errorMessage, setErrorMessage] = useState('');
  const navigate = useNavigate();

  const handleLogin = (e) => {
    e.preventDefault();

    // Dummy check: If email and password match dummy values
    if (email === 'test@estexample.com' && password === 'password123') {
      navigate('/viewUser');  // Redirect to the user list page
    } else {
      setErrorMessage('Invalid email or password');
    }
  };

  return (
```

```
nts\LAMBTON COLLEGE\3RD Term\Full stack software development\Final FullStack JavaScript MERN Project\C0910925-MERN_Final_Assignment>

PS C:\Users\Kisan Rai\OneDrive - Lambton College\Documents\LAMBTON COLLEGE\3RD Term\Full stack software development\Final FullStack JavaScript MERN Project\C0910925-ME
RN_Final_Assignment> cd backend
PS C:\Users\Kisan Rai\OneDrive - Lambton College\Documents\LAMBTON COLLEGE\3RD Term\Full stack software development\Final FullStack JavaScript MERN Project\C0910925-ME
RN_Final_Assignment\backend> npm start

> backend@1.0.0 start
> node app.js

Server running on port 5000
Connected to MongoDB
```

To check the backend server and view the user information, use the following URL to see all the populated user data in JSON format:

http://localhost:5000/api/users



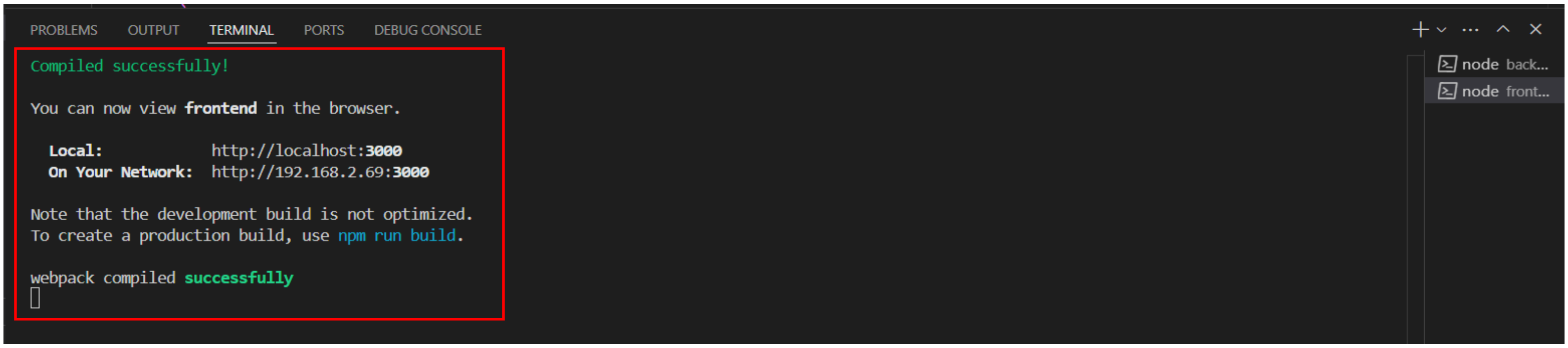
```
{
  "_id": "6746f5e4454518cc5c3258aa",
  "firstName": "Velma",
  "lastName": "Gusikowski",
  "dateOfBirth": "2024-11-26T02:23:23.494Z",
  "address1": "73903 Bell Lane",
  "address2": "Suite 624",
  "city": "Lake Betty",
  "postalCode": "33784-3416",
  "country": "Chile",
  "phoneNumber": "547-962-1711 x55972",
  "email": "Raven_Hammes@gmail.com",
  "userNotes": "Sopor degero comminor solutio argentum adfero.",
  "__v": 0
},
{
  "_id": "6746f5e5454518cc5c3258ad",
  "firstName": "Bernardo",
  "lastName": "Hyatt-Fadel",
  "dateOfBirth": "2024-10-31T00:00:00.000Z",
  "address1": "88443 Albert Road",
  "address2": "Apt. 193",
  "city": "North Enid",
  "postalCode": "37648-6251",
  "country": "Bangladesh",
  "phoneNumber": "794-977-2257 x3798",
  "email": "Jake.Kling@hotmail.com",
  "userNotes": "Terror videlicet antiquus dolore super concido totam vorax pariatur.",
  "__v": 0
},
{
  "_id": "6746f5e5454518cc5c3258bd",
  "firstName": "Roger",
  "lastName": "Rodriguez",
  "dateOfBirth": "2024-03-29T02:55:08.272Z",
  "address1": "61245 Greenwood Road",
  "address2": "Apt. 756",
  "city": "Marcellusside",
  "postalCode": "86112",
  "country": "Mongolia",
  "phoneNumber": "(736) 763-1342 x51998",
  "email": "Sophia.Kuhic@yahoo.com",
  "userNotes": "Curiositas cuppedia possimus amet adhuc.",
  "__v": 0
},
{
  "_id": "6747bfae79c6ed9aaed0e605",
  "firstName": "Kisan",
  "lastName": "Rai",
  "dateOfBirth": "1999-08-12T00:00:00.000Z",
  "address1": "sarnia",
  "address2": "192 brock stn",
  "city": "sarnia",
  "postalCode": "N7s 53Z",
  "country": "Canada",
  "phoneNumber": "4376610038",
  "email": "kisanrai939@gmail.com",
  "userNotes": "Testing",
  "__v": 0
```

As we can see the user information in JSON format from the backend server in the screenshot above, you can view specific user information by using the user ID. For example, to see the details of a specific user, use the following URL:

http://localhost:5000/api/users/674900dd0f375e1fbe7aea90

10.2 Fronted server

To run the frontend server, open another terminal and navigate to the frontend directory by typing **[cd fronted]** and hitting Enter. This will change the directory to the front-end folder. Then, type **[npm start]** to run the fronted reactJs server



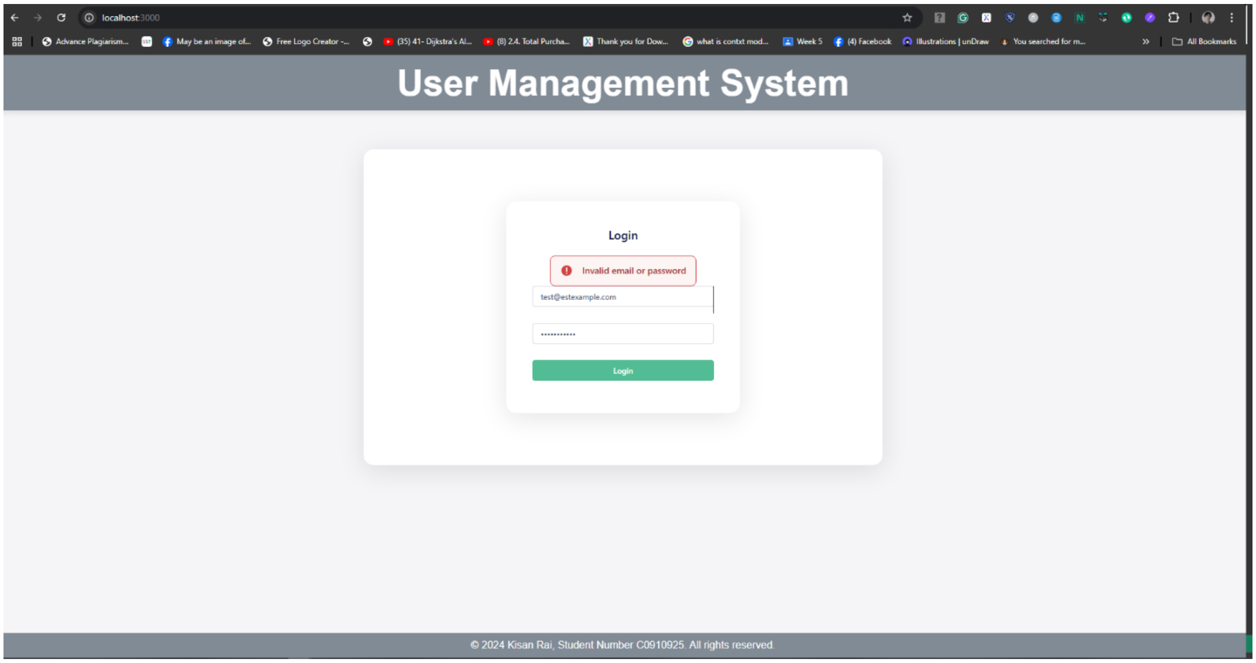
```
Compiled successfully!

You can now view frontend in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.2.69:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

It will open in your browser at **http://localhost:3000/**, which is the default login page of my application.

11. Login Page

Use the dummy email: **test@estexample.com** and password: **password123** for login credentials. If the credentials are incorrect, an error message will pop up. If the login is successful, it will redirect you to the "View User" page.

For a invalid login



localhost:3000

User Management System

Login

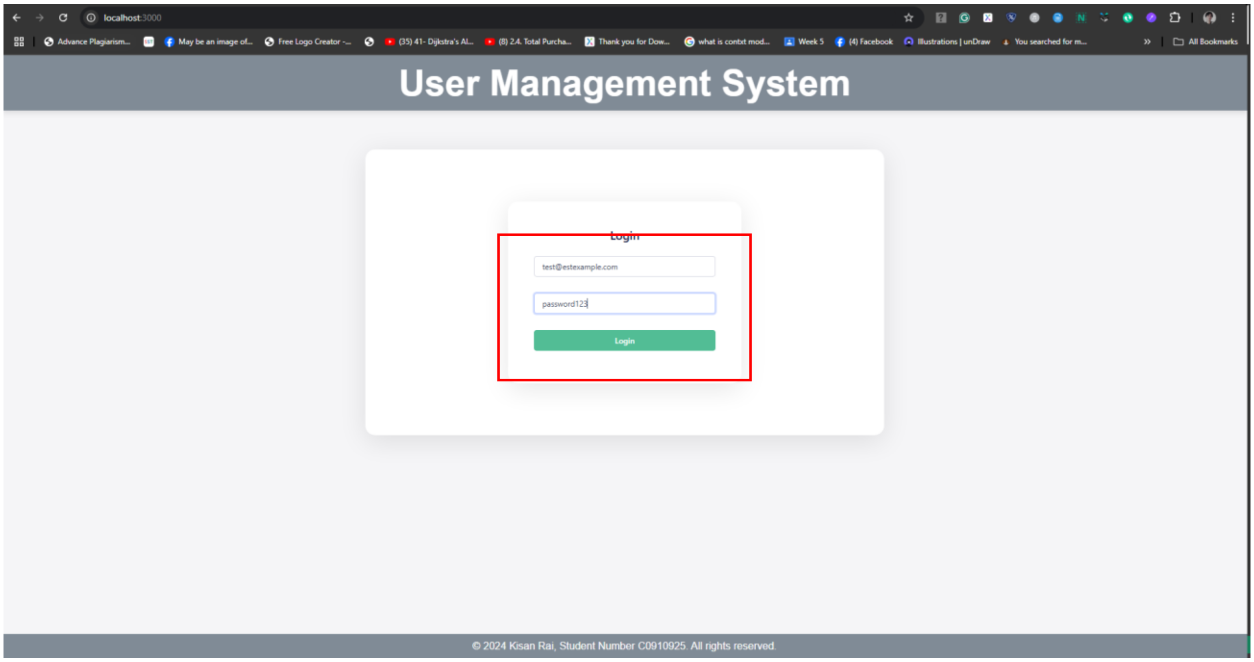
Invalid email or password

test@estexample.com

•••••••••••

Login

© 2024 Kisan Rai, Student Number C0910925. All rights reserved.



localhost:3000

User Management System

Login

test@estexample.com

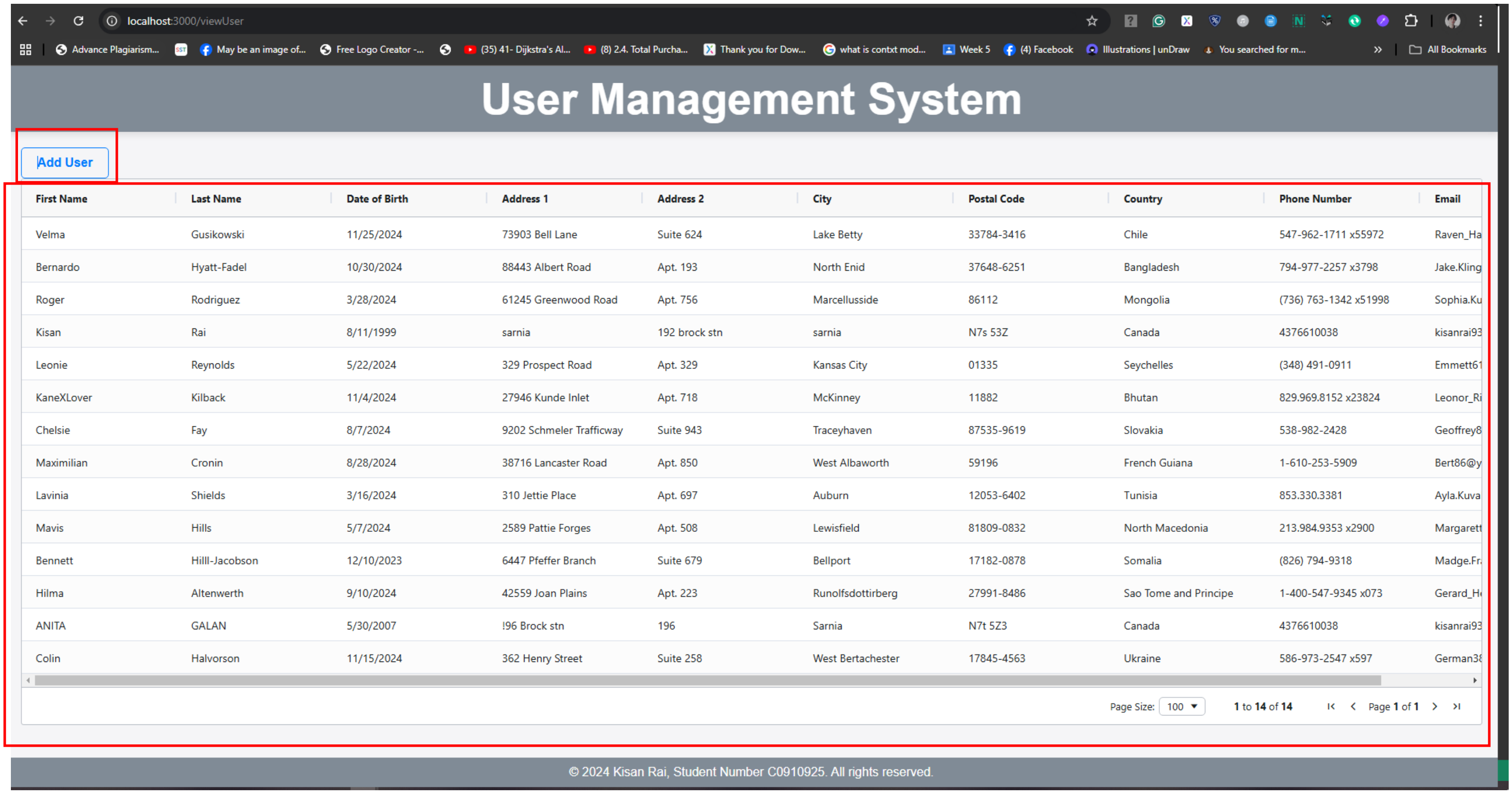
password123

Login

© 2024 Kisan Rai, Student Number C0910925. All rights reserved.

12. View User page

Once the login is successful, you will be redirected to the main page, which is the "View User" page. On this page, you will be able to see all the user information displayed in an AG Grid. The data is fetched from MongoDB and shown in the AG Grid on the "View User" page.



User Management System

Add User

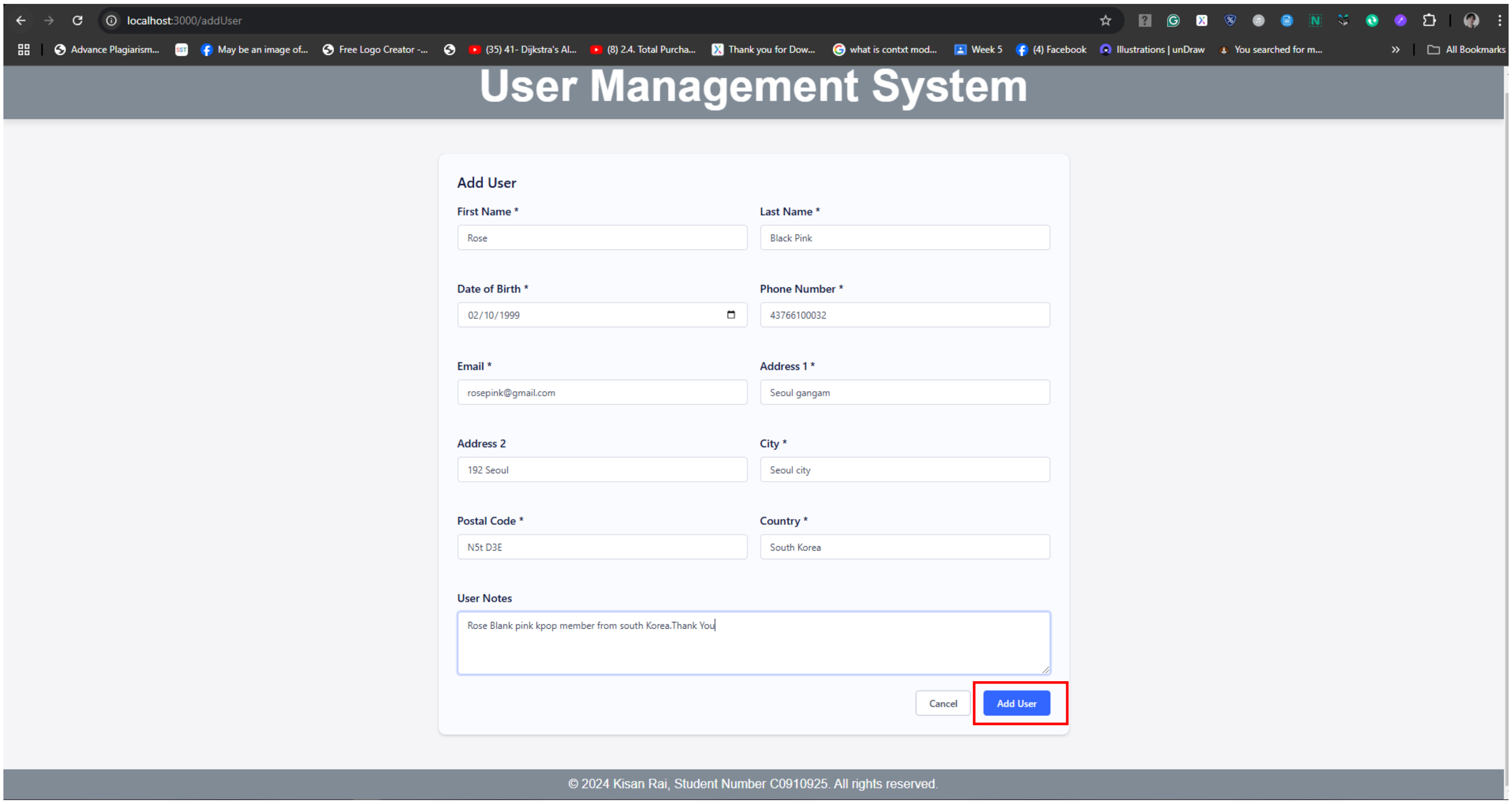
First Name	Last Name	Date of Birth	Address 1	Address 2	City	Postal Code	Country	Phone Number	Email
Velma	Gusikowski	11/25/2024	73903 Bell Lane	Suite 624	Lake Betty	33784-3416	Chile	547-962-1711 x55972	Raven_Ha
Bernardo	Hyatt-Fadel	10/30/2024	88443 Albert Road	Apt. 193	North Enid	37648-6251	Bangladesh	794-977-2257 x3798	Jake.Kling
Roger	Rodriguez	3/28/2024	61245 Greenwood Road	Apt. 756	Marcellusside	86112	Mongolia	(736) 763-1342 x51998	Sophia.Ku
Kisan	Rai	8/11/1999	sarnia	192 brock stn	sarnia	N7s 53Z	Canada	4376610038	kisanrai93
Leonie	Reynolds	5/22/2024	329 Prospect Road	Apt. 329	Kansas City	01335	Seychelles	(348) 491-0911	Emmett61
KaneXLover	Kilback	11/4/2024	27946 Kunde Inlet	Apt. 718	McKinney	11882	Bhutan	829.969.8152 x23824	Leonor_Ri
Chelsie	Fay	8/7/2024	9202 Schmeler Trafficway	Suite 943	Traceyhaven	87535-9619	Slovakia	538-982-2428	Geoffrey8
Maximilian	Cronin	8/28/2024	38716 Lancaster Road	Apt. 850	West Albaworth	59196	French Guiana	1-610-253-5909	Bert86@y
Lavinia	Shields	3/16/2024	310 Jettie Place	Apt. 697	Auburn	12053-6402	Tunisia	853.330.3381	Ayla.Kuva
Mavis	Hills	5/7/2024	2589 Pattie Forges	Apt. 508	Lewisfield	81809-0832	North Macedonia	213.984.9353 x2900	Margarett
Bennett	Hilll-Jacobson	12/10/2023	6447 Pfeffer Branch	Suite 679	Bellport	17182-0878	Somalia	(826) 794-9318	Madge.Fr
Hilma	Altenwerth	9/10/2024	42559 Joan Plains	Apt. 223	Runolfsdottirberg	27991-8486	Sao Tome and Principe	1-400-547-9345 x073	Gerard_H
ANITA	GALAN	5/30/2007	!96 Brock stn	196	Sarnia	N7t 5Z3	Canada	4376610038	kisanrai93
Colin	Halvorson	11/15/2024	362 Henry Street	Suite 258	West Bertachester	17845-4563	Ukraine	586-973-2547 x597	German3

Page Size: 100 ▾ 1 to 14 of 14 |‹ ‹ Page 1 of 1 › ›|

© 2024 Kisan Rai, Student Number C0910925. All rights reserved.

13. Add user page

Here, I am adding user information for a user named **Rose** on the "Add User" page.



User Management System

Add User

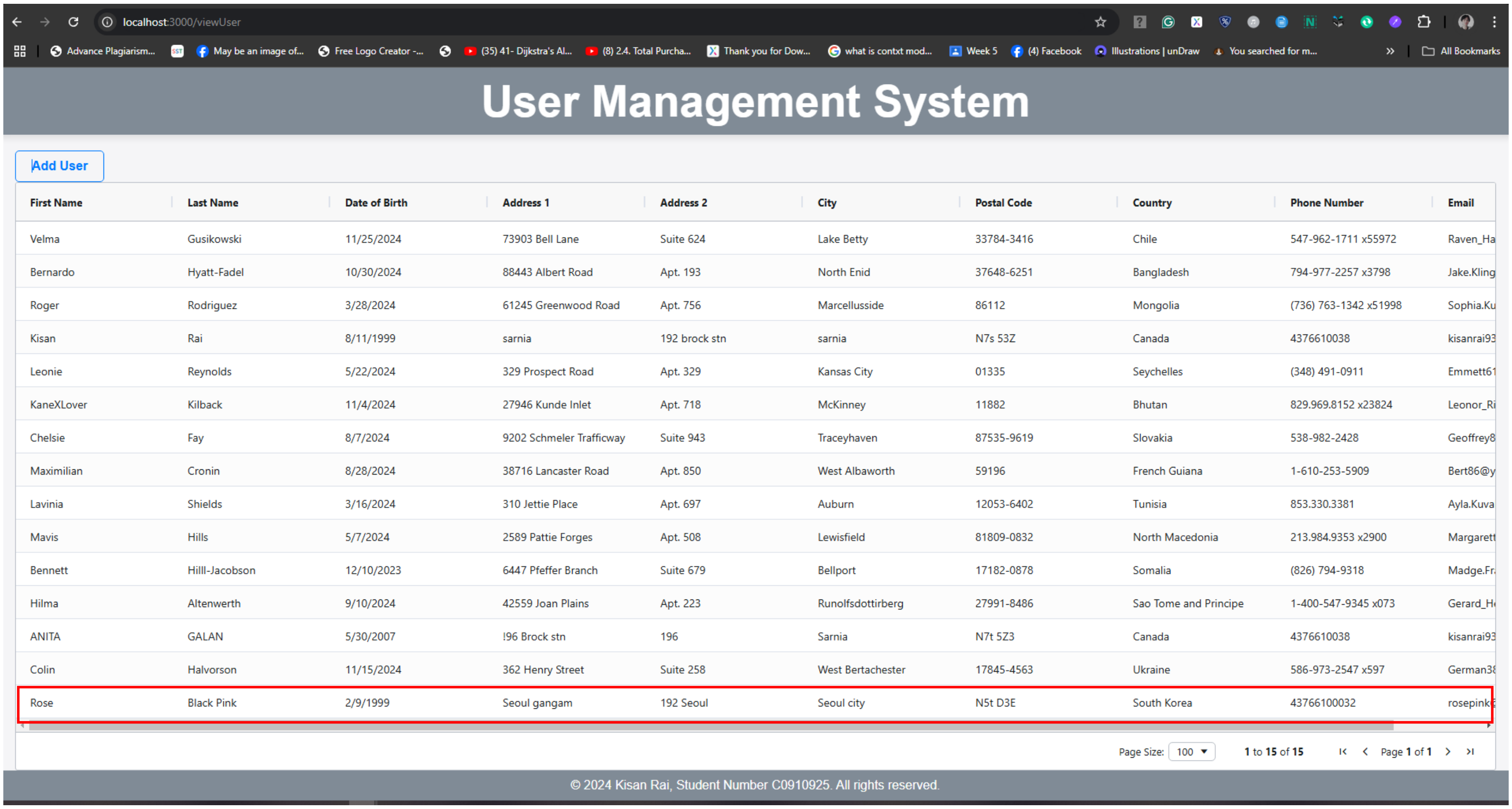
First Name *	Last Name *
Rose	Black Pink
Date of Birth *	Phone Number *
02/10/1999	43766100032
Email *	Address 1 *
rosepink@gmail.com	Seoul gangam
Address 2	City *
192 Seoul	Seoul city
Postal Code *	Country *
N5t D3E	South Korea

User Notes

Rose Blank pink kpop member from south Korea.Thank You

© 2024 Kisan Rai, Student Number C0910925. All rights reserved.

14. User is added into view user AG grid page.



User Management System

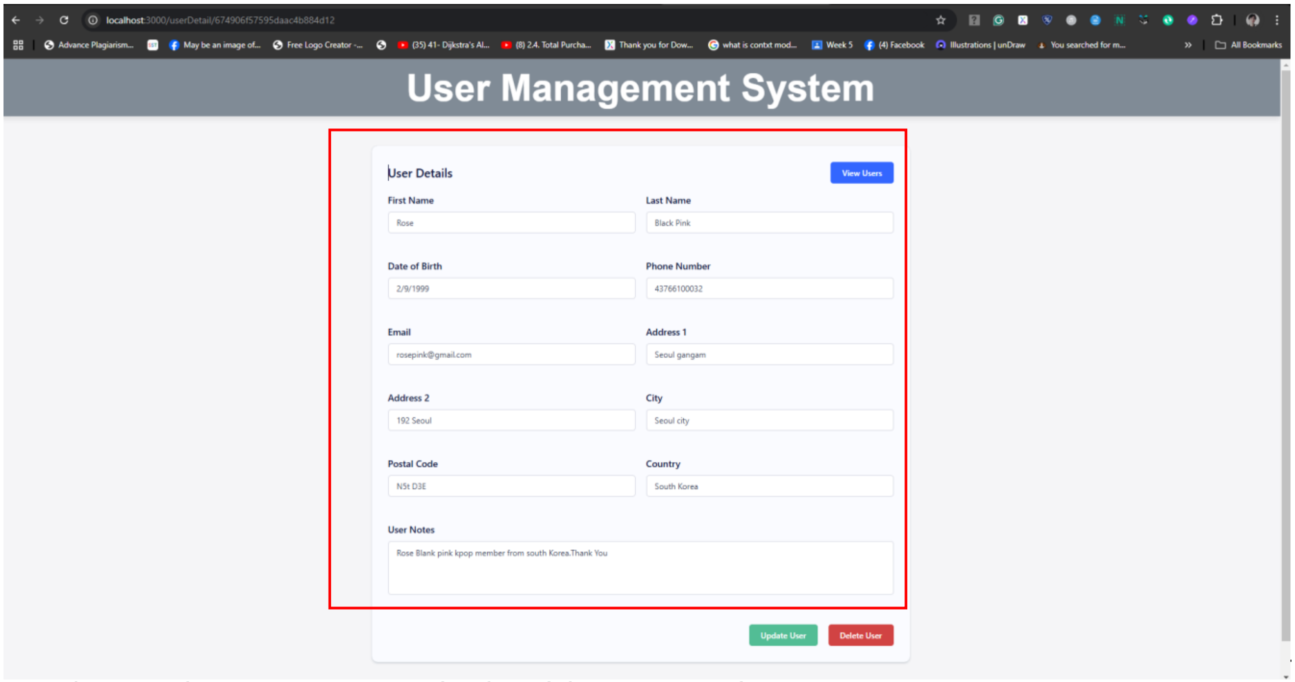
Add User

First Name	Last Name	Date of Birth	Address 1	Address 2	City	Postal Code	Country	Phone Number	Email
Velma	Gusikowski	11/25/2024	73903 Bell Lane	Suite 624	Lake Betty	33784-3416	Chile	547-962-1711 x55972	Raven_Ha
Bernardo	Hyatt-Fadel	10/30/2024	88443 Albert Road	Apt. 193	North Enid	37648-6251	Bangladesh	794-977-2257 x3798	Jake.Kling
Roger	Rodriguez	3/28/2024	61245 Greenwood Road	Apt. 756	Marcellusside	86112	Mongolia	(736) 763-1342 x51998	Sophia.Ku
Kisan	Rai	8/11/1999	sarnia	192 brock stn	sarnia	N7s 53Z	Canada	4376610038	kisanrai93
Leonie	Reynolds	5/22/2024	329 Prospect Road	Apt. 329	Kansas City	01335	Seychelles	(348) 491-0911	Emmett61
KaneXLover	Kilback	11/4/2024	27946 Kunde Inlet	Apt. 718	McKinney	11882	Bhutan	829.969.8152 x23824	Leonor_Ri
Chelsie	Fay	8/7/2024	9202 Schmeler Trafficway	Suite 943	Traceyhaven	87535-9619	Slovakia	538-982-2428	Geoffrey8
Maximilian	Cronin	8/28/2024	38716 Lancaster Road	Apt. 850	West Albaworth	59196	French Guiana	1-610-253-5909	Bert86@y
Lavinia	Shields	3/16/2024	310 Jettie Place	Apt. 697	Auburn	12053-6402	Tunisia	853.330.3381	Ayla.Kuva
Mavis	Hills	5/7/2024	2589 Pattie Forges	Apt. 508	Lewisfield	81809-0832	North Macedonia	213.984.9353 x2900	Margarett
Bennett	Hilll-Jacobson	12/10/2023	6447 Pfeffer Branch	Suite 679	Bellport	17182-0878	Somalia	(826) 794-9318	Madge.Fr
Hilma	Altenwerth	9/10/2024	42559 Joan Plains	Apt. 223	Runolfsdottirberg	27991-8486	Sao Tome and Principe	1-400-547-9345 x073	Gerard_H
ANITA	GALAN	5/30/2007	!96 Brock stn	196	Sarnia	N7t 5Z3	Canada	4376610038	kisanrai93
Colin	Halvorson	11/15/2024	362 Henry Street	Suite 258	West Bertachester	17845-4563	Ukraine	586-973-2547 x597	German38
Rose	Black Pink	2/9/1999	Seoul gangam	192 Seoul	Seoul city	N5t D3E	South Korea	43766100032	rosepink

Page Size: 100 ▾ 1 to 15 of 15 |‹ ‹ Page 1 of 1 › ›|

© 2024 Kisan Rai, Student Number C0910925. All rights reserved.

15. Rose User Detail information page.



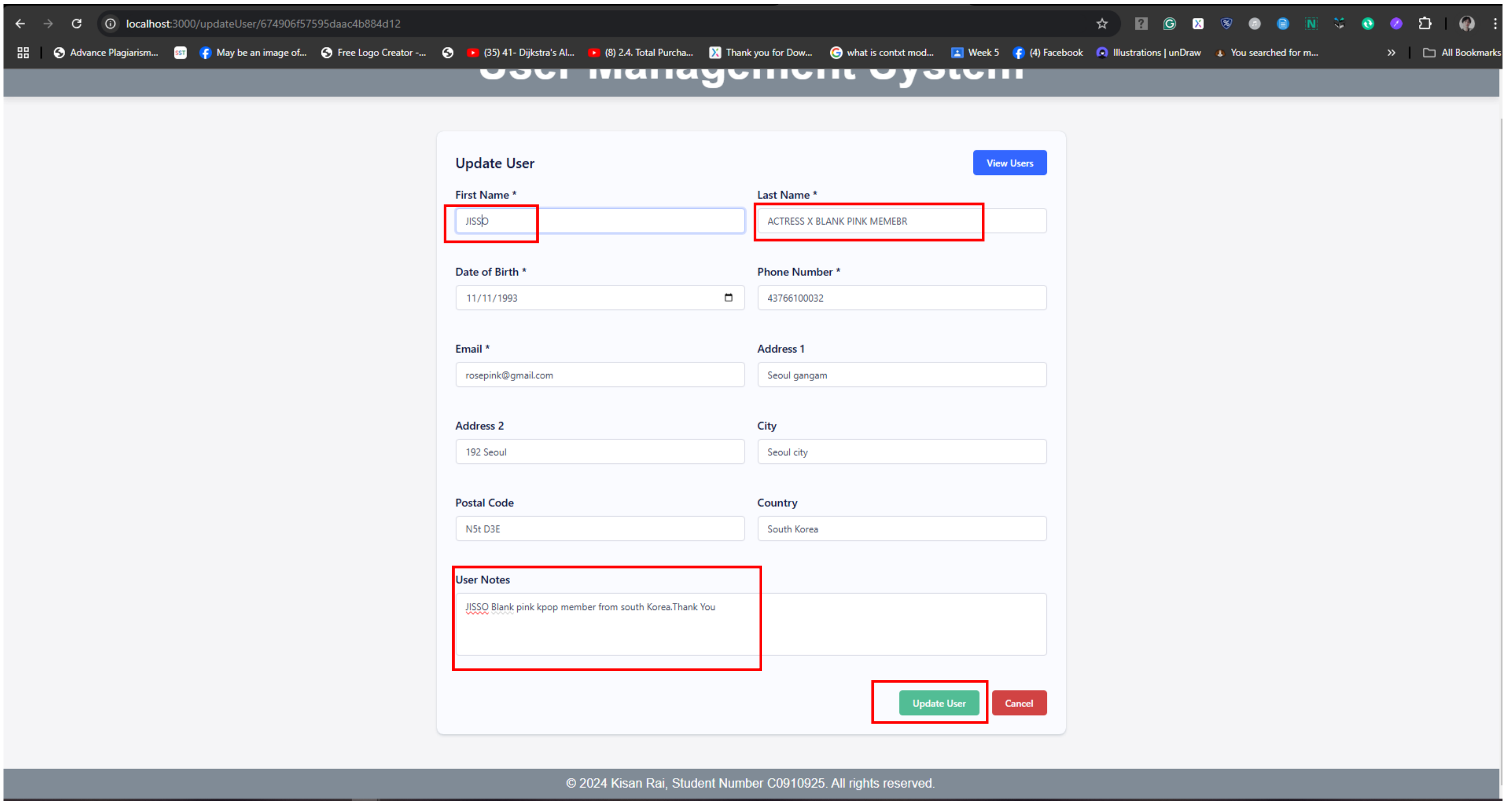
User Management System

User Details

First Name: Rose	Last Name: Black Pink
Date of Birth: 2/9/1999	Phone Number: 43766100032
Email: rosepink@gmail.com	Address 1: Seoul gangam
Address 2: 192 Seoul	City: Seoul city
Postal Code: N5t D3E	Country: South Korea

User Notes: Rose Blank pink kpop member from south Korea.Thank You

16. Updating Rose’s information in updateUser page



localhost:3000/updateUser/674906f57595daac4b884d12

Update User

First Name *
JISSO

Last Name *
ACTRESS X BLANK PINK MEMEBR

Date of Birth *
11/11/1993

Phone Number *
43766100032

Email *
rosepink@gmail.com

Address 1
Seoul gangam

Address 2
192 Seoul

City
Seoul city

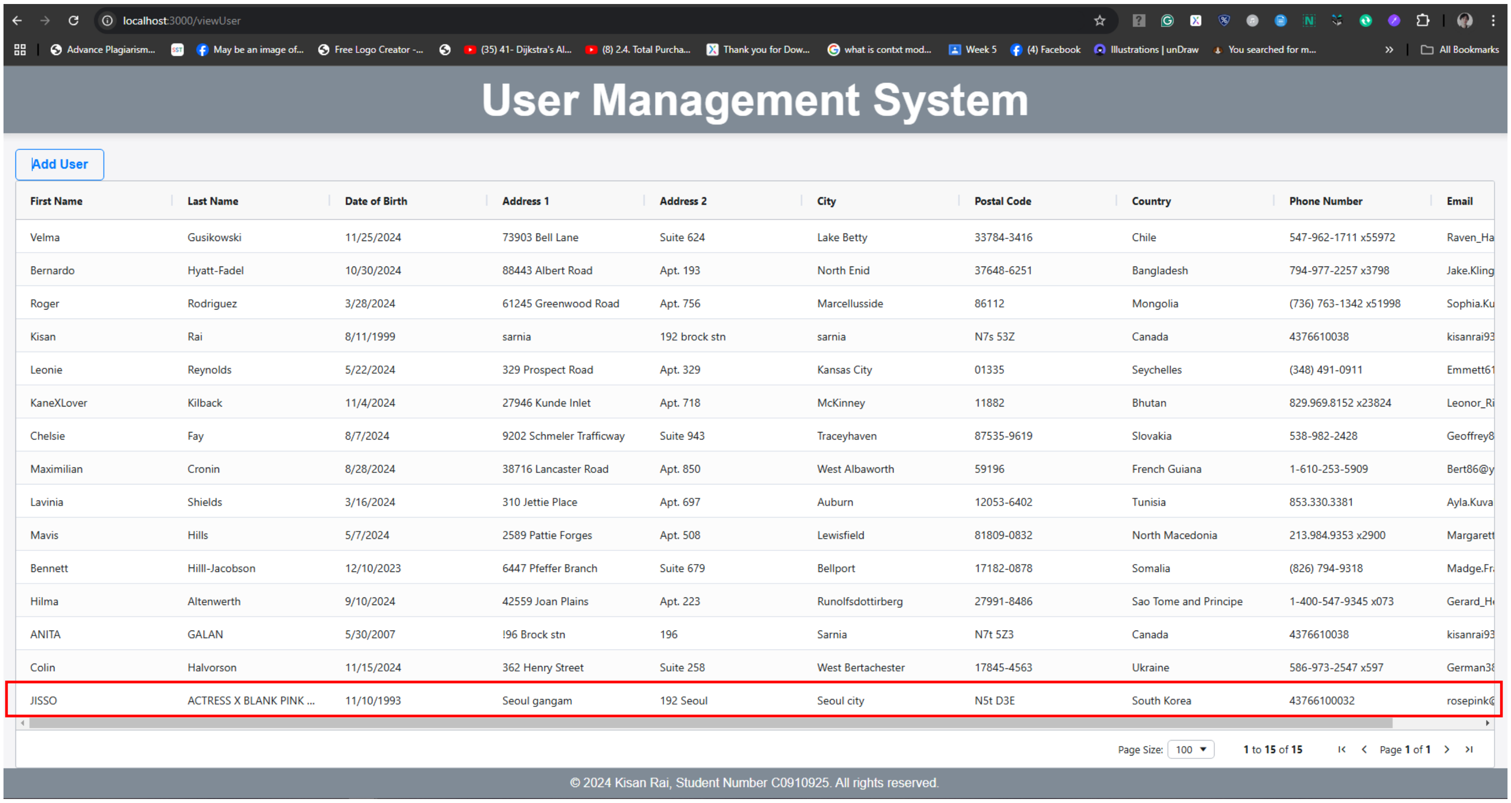
Postal Code
N5t D3E

Country
South Korea

User Notes
JISSO Blank pink kpop member from south Korea.Thank You

© 2024 Kisan Rai, Student Number C0910925. All rights reserved.

17. user information updated from Rose to JISSO information in viewUser page inside the AG grid.



User Management System

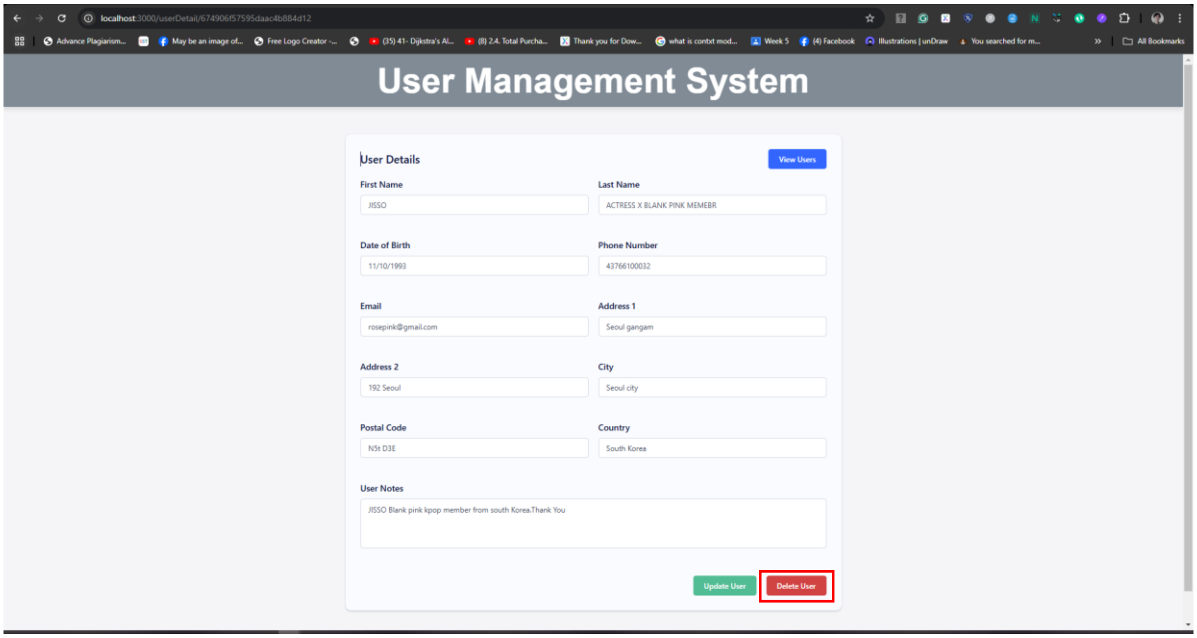
Add User

First Name	Last Name	Date of Birth	Address 1	Address 2	City	Postal Code	Country	Phone Number	Email
Velma	Gusikowski	11/25/2024	73903 Bell Lane	Suite 624	Lake Betty	33784-3416	Chile	547-962-1711 x55972	Raven_Ha
Bernardo	Hyatt-Fadel	10/30/2024	88443 Albert Road	Apt. 193	North Enid	37648-6251	Bangladesh	794-977-2257 x3798	Jake.Kling
Roger	Rodriguez	3/28/2024	61245 Greenwood Road	Apt. 756	Marcellusside	86112	Mongolia	(736) 763-1342 x51998	Sophia.Ku
Kisan	Rai	8/11/1999	sarnia	192 brock stn	sarnia	N7s 53Z	Canada	4376610038	kisanrai93
Leonie	Reynolds	5/22/2024	329 Prospect Road	Apt. 329	Kansas City	01335	Seychelles	(348) 491-0911	Emmett61
KaneXLover	Kilback	11/4/2024	27946 Kunde Inlet	Apt. 718	McKinney	11882	Bhutan	829.969.8152 x23824	Leonor_Ri
Chelsie	Fay	8/7/2024	9202 Schmeler Trafficway	Suite 943	Traceyhaven	87535-9619	Slovakia	538-982-2428	Geoffrey8
Maximilian	Cronin	8/28/2024	38716 Lancaster Road	Apt. 850	West Albaworth	59196	French Guiana	1-610-253-5909	Bert86@y
Lavinia	Shields	3/16/2024	310 Jettie Place	Apt. 697	Auburn	12053-6402	Tunisia	853.330.3381	Ayla.Kuva
Mavis	Hills	5/7/2024	2589 Pattie Forges	Apt. 508	Lewisfield	81809-0832	North Macedonia	213.984.9353 x2900	Margarett
Bennett	Hilll-Jacobson	12/10/2023	6447 Pfeffer Branch	Suite 679	Bellport	17182-0878	Somalia	(826) 794-9318	Madge.Fr
Hilma	Altenwerth	9/10/2024	42559 Joan Plains	Apt. 223	Runolfsdottirberg	27991-8486	Sao Tome and Principe	1-400-547-9345 x073	Gerard_H
ANITA	GALAN	5/30/2007	!96 Brock stn	196	Sarnia	N7t 5Z3	Canada	4376610038	kisanrai93
Colin	Halvorson	11/15/2024	362 Henry Street	Suite 258	West Bertachester	17845-4563	Ukraine	586-973-2547 x597	German38
JISSO	ACTRESS X BLANK PINK ...	11/10/1993	Seoul gangam	192 Seoul	Seoul city	N5t D3E	South Korea	43766100032	rosepink@

Page Size: 100 ▾ 1 to 15 of 15 Page 1 of 1

© 2024 Kisan Rai, Student Number C0910925. All rights reserved.

18. Deleting user information



User Management System

User Details

View Users

First Name: JISSO

Last Name: ACTRESS X BLANK PINK MEMEBR

Date of Birth: 11/10/1993

Phone Number: 43766100032

Email: rosepink@gmail.com

Address 1: Seoul gangam

Address 2: 192 Seoul

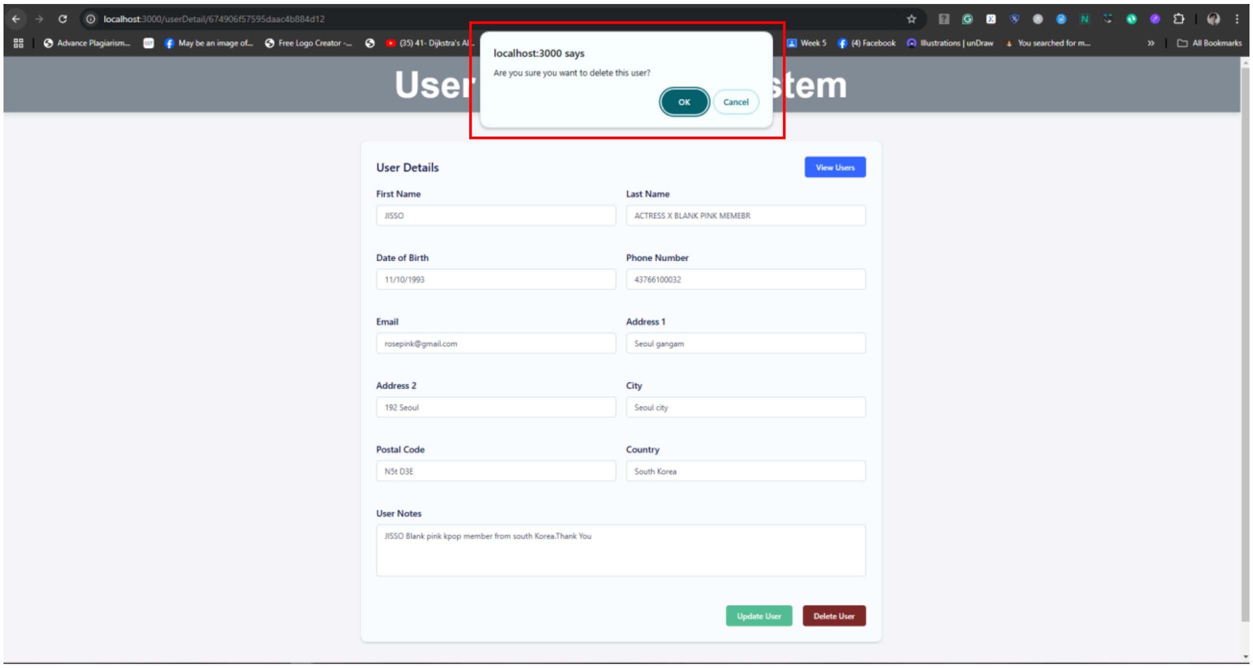
City: Seoul city

Postal Code: N5t D3E

Country: South Korea

User Notes: JISSO Blank pink kpop member from south Korea.Thank You

Update User Delete User



localhost:3000 says

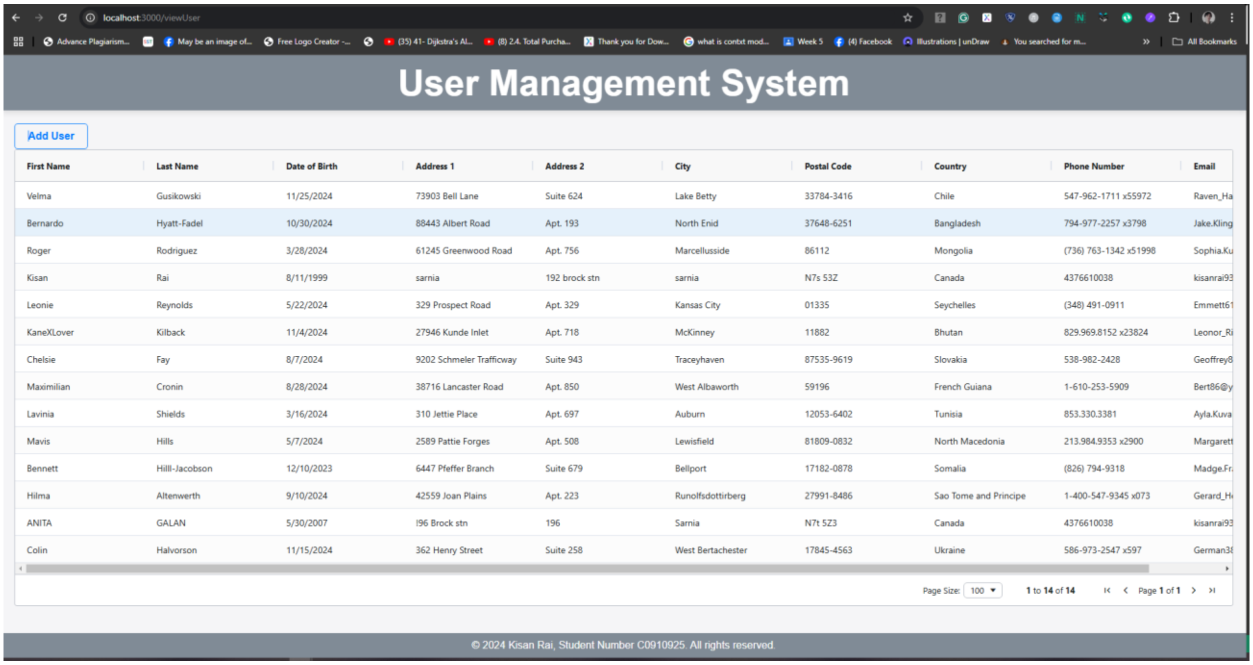
Are you sure you want to delete this user?

User Details

First Name	Last Name
JISSO	ACTRESS X BLANK PINK MEMEBR
Date of Birth	Phone Number
11/10/1993	43766100032
Email	Address 1
rosepink@gmail.com	Seoul gangam
Address 2	City
192 Seoul	Seoul city
Postal Code	Country
N5t D3E	South Korea

User Notes

JISSO Blank pink kpop member from south Korea.Thank You



User Management System

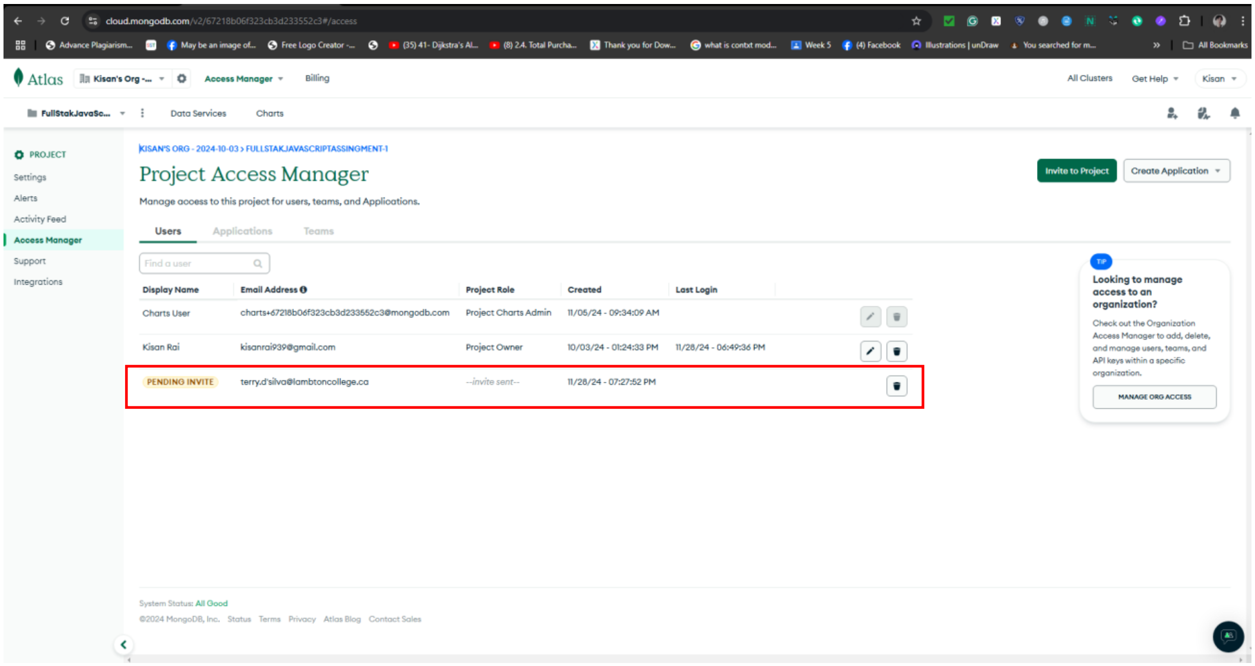
First Name	Last Name	Date of Birth	Address 1	Address 2	City	Postal Code	Country	Phone Number	Email
Velma	Gusikowski	11/25/2024	73903 Bell Lane	Suite 624	Lake Betty	33784-3416	Chile	547-962-1711 x55972	Raven_Ha
Bernardo	Hyatt-Fadel	10/30/2024	88443 Albert Road	Apt. 193	North Enid	37648-6251	Bangladesh	794-977-2257 x3798	Jake.Kling
Roger	Rodriguez	3/28/2024	61245 Greenwood Road	Apt. 756	Marcellusside	86112	Mongolia	(736) 763-1342 x51998	Sophia.Ku
Kisan	Rai	8/11/1999	sarnia	192 brock stn	sarnia	N7s 53Z	Canada	4376610038	kisanrai93
Leonie	Reynolds	5/22/2024	329 Prospect Road	Apt. 329	Kansas City	01335	Seychelles	(348) 491-0911	Emmett61
KaneXLover	Kilback	11/4/2024	27946 Kunde Inlet	Apt. 718	McKinney	11882	Bhutan	829.969.8152 x23824	Leonor_Ri
Chelsie	Fay	8/7/2024	9202 Schmeler Trafficway	Suite 943	Traceyhaven	87535-9619	Slovakia	538-982-2428	Geoffrey8
Maximilian	Cronin	8/28/2024	38716 Lancaster Road	Apt. 850	West Albaworth	59196	French Guiana	1-610-253-5909	Bert86@y
Lavinia	Shields	3/16/2024	310 Jettie Place	Apt. 697	Auburn	12053-6402	Tunisia	853.330.3381	Ayla.Kuva
Mavis	Hills	5/7/2024	2589 Pattie Forges	Apt. 508	Lewisfield	81809-0832	North Macedonia	213.984.9353 x2900	Margarett
Bennett	Hilll-Jacobson	12/10/2023	6447 Pfeffer Branch	Suite 679	Bellport	17182-0878	Somalia	(826) 794-9318	Madge.Fr
Hilma	Altenwerth	9/10/2024	42559 Joan Plains	Apt. 223	Runolfsdottirberg	27991-8486	Sao Tome and Principe	1-400-547-9345 x073	Gerard_H
ANITA	GALAN	5/30/2007	I96 Brock stn	196	Sarnia	N7t 5Z3	Canada	4376610038	kisanrai93
Colin	Halvorson	11/15/2024	362 Henry Street	Suite 258	West Bertachester	17845-4563	Ukraine	586-973-2547 x597	German38

Page Size: 100 ▾ 1 to 14 of 14 |‹ ‹ Page 1 of 1 › ›|

© 2024 Kisan Rai, Student Number C0910925. All rights reserved.

19. MongoDB Atlas

I have shared my MongoDB project with you. Please let me know if you are able to access my project database, as you can see in the highlighted red box in the screenshot above.



KISAN'S ORG - 2024-10-03 > FULLSTAKJAVASCRIPTASSINGMENT-1

Project Access Manager

Manage access to this project for users, teams, and Applications.

Users | Applications | Teams

Display Name	Email Address	Project Role	Created	Last Login
Charts User	charts+67218b06f323cb3d233552c3@mongodb.com	Project Charts Admin	11/05/24 - 09:34:09 AM	
Kisan Rai	kisanrai939@gmail.com	Project Owner	10/03/24 - 01:24:33 PM	11/28/24 - 06:49:36 PM
PENDING INVITE	terry.d'silva@lambtoncollege.ca	--invite sent--	11/28/24 - 07:27:52 PM	

TIP

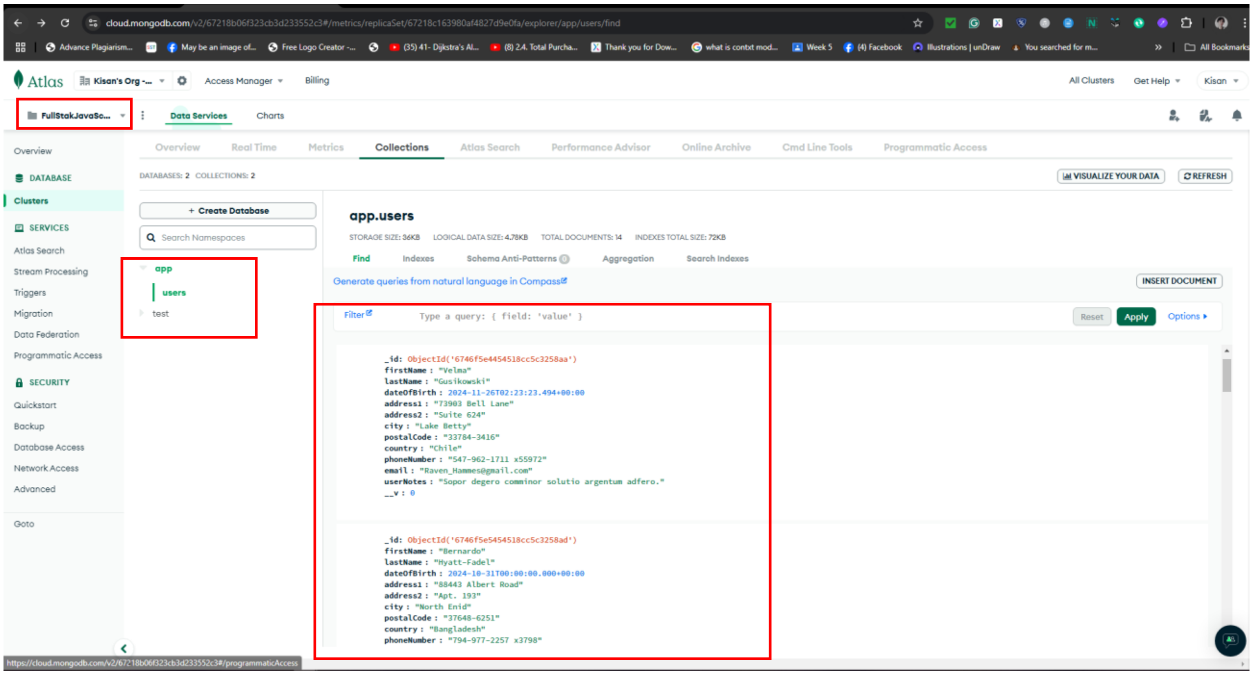
Looking to manage access to an organization?

Check out the Organization Access Manager to add, delete, and manage users, teams, and API keys within a specific organization.

MANAGE ORG ACCESS

System Status: All Good

©2024 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales



DATABASES: 2 COLLECTIONS: 2

app.users

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 4.78KB TOTAL DOCUMENTS: 14 INDEXES TOTAL SIZE: 72KB

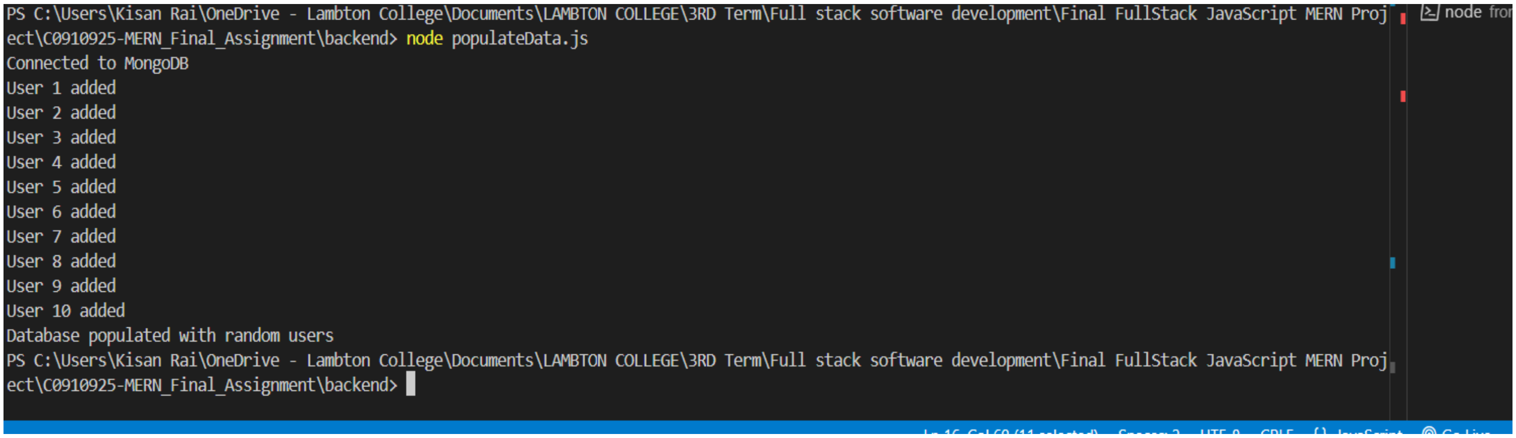
Generate queries from natural language in Compass

Filter Type a query: { field: 'value' }

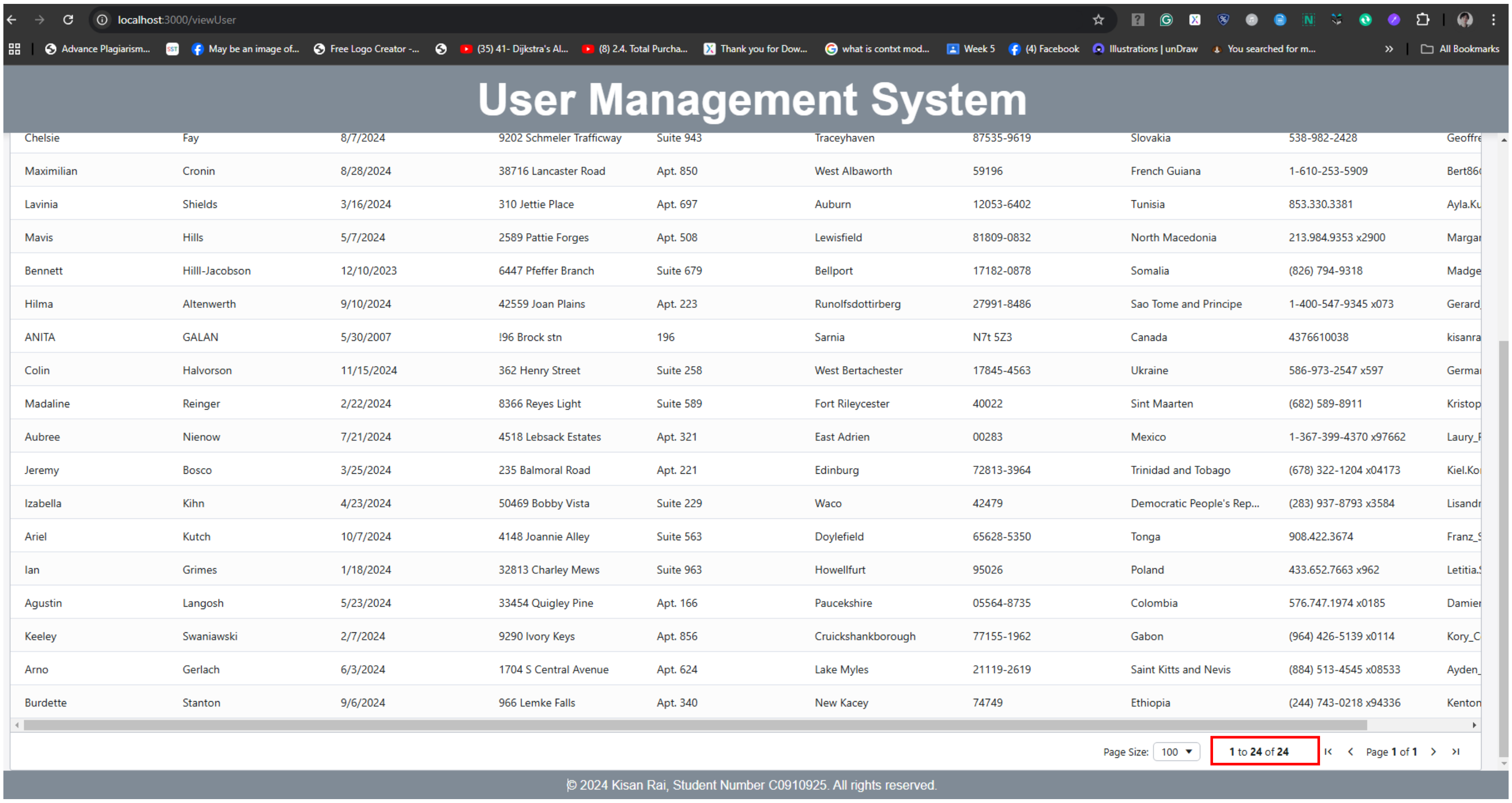
```
_id: ObjectId('6746f5e4454518cc5c3258aa')
firstName : "Velma"
lastName : "Gusikowski"
dateOfBirth : 2024-11-26T02:23:23.494+00:00
address1 : "73903 Bell Lane"
address2 : "Suite 624"
city : "Lake Betty"
postalCode : "33784-3416"
country : "Chile"
phoneNumber : "547-962-1711 x55972"
email : "Raven_Hammes@gmail.com"
userNotes : "Sopor degero comminor solutio argentum adfero."
__v : 0

_id: ObjectId('6746f5e5454518cc5c3258ad')
firstName : "Bernardo"
lastName : "Hyatt-Fadel"
dateOfBirth : 2024-10-31T00:00:00.000+00:00
address1 : "88443 Albert Road"
address2 : "Apt. 193"
city : "North Enid"
postalCode : "37648-6251"
country : "Bangladesh"
phoneNumber : "794-977-2257 x3798"
```

20. Populating data

I have populated 10 random user data into the backend server, which were added as 10 users. This data was fetched into MongoDB Atlas and is now reflected on the "View User" page in the AG Grid.



```
PS C:\Users\Kisan Rai\OneDrive - Lambton College\Documents\LAMBTON COLLEGE\3RD Term\Full stack software development\Final FullStack JavaScript MERN Proj
ect\C0910925-MERN_Final_Assignment\backend> node populateData.js
Connected to MongoDB
User 1 added
User 2 added
User 3 added
User 4 added
User 5 added
User 6 added
User 7 added
User 8 added
User 9 added
User 10 added
Database populated with random users
PS C:\Users\Kisan Rai\OneDrive - Lambton College\Documents\LAMBTON COLLEGE\3RD Term\Full stack software development\Final FullStack JavaScript MERN Proj
ect\C0910925-MERN_Final_Assignment\backend> 
```



User Management System

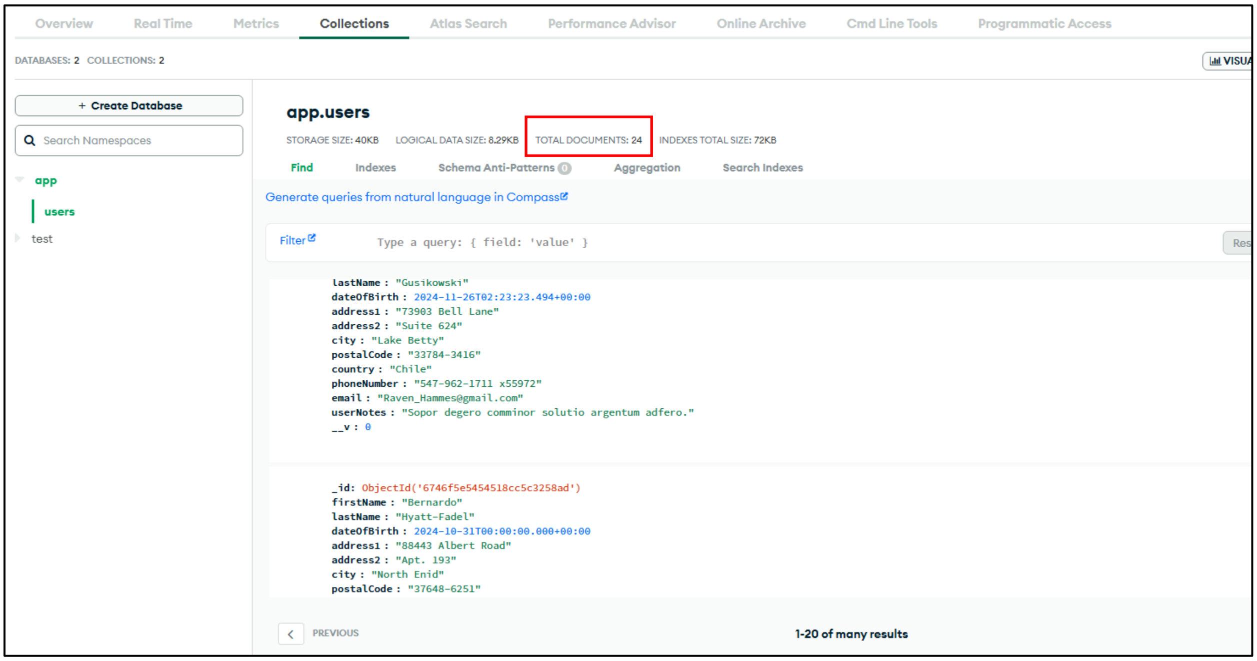
Chelsie	Fay	8/7/2024	9202 Schmeler Trafficway	Suite 943	Traceyhaven	87535-9619	Slovakia	538-982-2428	Geoffre
Maximilian	Cronin	8/28/2024	38716 Lancaster Road	Apt. 850	West Albaworth	59196	French Guiana	1-610-253-5909	Bert86
Lavinia	Shields	3/16/2024	310 Jettie Place	Apt. 697	Auburn	12053-6402	Tunisia	853.330.3381	Ayla.Ku
Mavis	Hills	5/7/2024	2589 Pattie Forges	Apt. 508	Lewisfield	81809-0832	North Macedonia	213.984.9353 x2900	Margar
Bennett	Hilll-Jacobson	12/10/2023	6447 Pfeffer Branch	Suite 679	Bellport	17182-0878	Somalia	(826) 794-9318	Madge
Hilma	Altenwerth	9/10/2024	42559 Joan Plains	Apt. 223	Runolfsdottirberg	27991-8486	Sao Tome and Principe	1-400-547-9345 x073	Gerard
ANITA	GALAN	5/30/2007	!96 Brock stn	196	Sarnia	N7t 5Z3	Canada	4376610038	kisanra
Colin	Halvorson	11/15/2024	362 Henry Street	Suite 258	West Bertachester	17845-4563	Ukraine	586-973-2547 x597	Germa
Madaline	Reinger	2/22/2024	8366 Reyes Light	Suite 589	Fort Rileycester	40022	Sint Maarten	(682) 589-8911	Kristop
Aubree	Nienow	7/21/2024	4518 Lebsack Estates	Apt. 321	East Adrien	00283	Mexico	1-367-399-4370 x97662	Laury_F
Jeremy	Bosco	3/25/2024	235 Balmoral Road	Apt. 221	Edinburg	72813-3964	Trinidad and Tobago	(678) 322-1204 x04173	Kiel.Ko
Izabella	Kihn	4/23/2024	50469 Bobby Vista	Suite 229	Waco	42479	Democratic People's Rep...	(283) 937-8793 x3584	Lisandr
Ariel	Kutch	10/7/2024	4148 Joannie Alley	Suite 563	Doylefield	65628-5350	Tonga	908.422.3674	Franz_S
Ian	Grimes	1/18/2024	32813 Charley Mews	Suite 963	Howellfurt	95026	Poland	433.652.7663 x962	Letitia.S
Agustin	Langosh	5/23/2024	33454 Quigley Pine	Apt. 166	Paucekshire	05564-8735	Colombia	576.747.1974 x0185	Damie
Keeley	Swaniawski	2/7/2024	9290 Ivory Keys	Apt. 856	Cruickshankborough	77155-1962	Gabon	(964) 426-5139 x0114	Kory_C
Arno	Gerlach	6/3/2024	1704 S Central Avenue	Apt. 624	Lake Myles	21119-2619	Saint Kitts and Nevis	(884) 513-4545 x08533	Ayden_
Burdette	Stanton	9/6/2024	966 Lemke Falls	Apt. 340	New Kacey	74749	Ethiopia	(244) 743-0218 x94336	Kenton

Page Size: 100 ▼ 1 to 24 of 24 Page 1 of 1

© 2024 Kisan Rai, Student Number C0910925. All rights reserved.

There were 14 records before, and after populating 10 more records, the total number of records has increased to 24.

Also reflected into MongoDB Atlas



DATABASES: 2 COLLECTIONS: 2

app.users

STORAGE SIZE: 40KB LOGICAL DATA SIZE: 8.29KB TOTAL DOCUMENTS: 24 INDEXES TOTAL SIZE: 72KB

```
lastName : "Gusikowski"
dateOfBirth : 2024-11-26T02:23:23.494+00:00
address1 : "73903 Bell Lane"
address2 : "Suite 624"
city : "Lake Betty"
postalCode : "33784-3416"
country : "Chile"
phoneNumber : "547-962-1711 x55972"
email : "Raven_Hammes@gmail.com"
userNotes : "Sopor degero comminor solutio argentum adfero."
__v : 0
```

```
_id: ObjectId('6746f5e5454518cc5c3258ad')
firstName : "Bernardo"
lastName : "Hyatt-Fadel"
dateOfBirth : 2024-10-31T00:00:00.000+00:00
address1 : "88443 Albert Road"
address2 : "Apt. 193"
city : "North Enid"
postalCode : "37648-6251"
```

1-20 of many results