**Abstract**

The Thaali Center project represents an innovative software solution aimed at empowering businesses and communities through streamlined user, community, and menu management functionalities. Leveraging modern technologies such as React, Express, and MongoDB, the Thaali Center provides an intuitive platform designed to enhance operational efficiency, ensure data security, and promote user satisfaction.

The primary goal of the Thaali Center is to facilitate seamless management of users, communities, and menus within a robust and user-friendly environment. By integrating features like user account management, community operations, and menu administration, the system offers comprehensive tools for organizations and communities to organize and optimize their operations effectively.

Key components of the Thaali Center include:

* User Management: The system allows for effortless creation, updating, and deletion of user profiles, ensuring efficient user account management.
* Community Management: Community administrators can add, update, and maintain community data with ease, fostering better community engagement and organization.
* Menu Management: The Thaali Center supports advanced menu management functionalities, enabling users to create, update, and delete menus seamlessly, with special support for the Gujarati language
* Data Security: Security is a top priority, with robust measures in place to protect user, community, and menu information from unauthorized access.
* Multilingual Support: The system is designed to enhance accessibility by supporting interactions and menu management in multiple languages, including Gujarati.
* Continuous Improvement: Feedback mechanisms are integrated to gather insights from users and communities, driving ongoing system enhancements and improvements.

Through its user-centric design, the Thaali Center aims to optimize workflow efficiency, foster community collaboration, and promote cultural inclusivity. With a responsive and secure platform, organizations and communities can confidently manage their resources, operations, and engagements, contributing to sustainable growth and development.

**Project Description: Thaali Center**

The Thaali Center project is a comprehensive software solution developed to address the management needs of businesses and communities, specifically focusing on user, community, menu management, feedback requests, and complaint handling. This description outlines the project's purpose, requirements, key features, and the technologies utilized.

The Thaali Center aims to streamline operations and enhance productivity by providing a centralized platform for managing users, communities, menus, feedback requests, and complaints. It facilitates efficient data management, user engagement, and resolution of customer feedback, catering to the needs of organizations and community initiatives.

**Needs:**

User Management: Simplified user account management with features for adding, updating, and deleting user profiles.

Community Operations: Effective community management tools for creating, updating, and maintaining community data.

Menu Administration: Advanced menu management functionalities, including creation, updates, and multilingual support (specifically in Gujarati).

Feedback Requests: Integration of feedback mechanisms to gather insights and enhance system functionality.

Complaint Handling: Module for managing and resolving customer complaints efficiently.

**Key Features:**

User Management:

* Create, update, and delete user profiles.
* Maintain user information securely.

Community Management:

* Add, update, and manage community data.
* Foster community engagement and organization.

Menu Management:

* Create, update, and delete menus effortlessly.
* Support for menu interactions in the Gujarati language.

Feedback Requests:

* Integrate feedback forms and mechanisms to gather user insights.
* Implement features for continuous system improvement based on feedback.

Complaint Handling:

* Provide a module for customers to submit complaints.
* Streamline complaint resolution processes and ensure timely responses.

Data Security:

* Implement robust security measures to protect user and community information.

Multilingual Support:

* Enhance accessibility by supporting menu interactions in multiple languages.

**Technologies and Tools Used:**

Frontend: React.js for building the user interface.

Backend: Express.js for server-side logic and API endpoints.

Database: MongoDB for data storage and management.

Additional Tools: Axios for HTTP requests, Bootstrap for responsive design, and various npm packages for project dependencies.

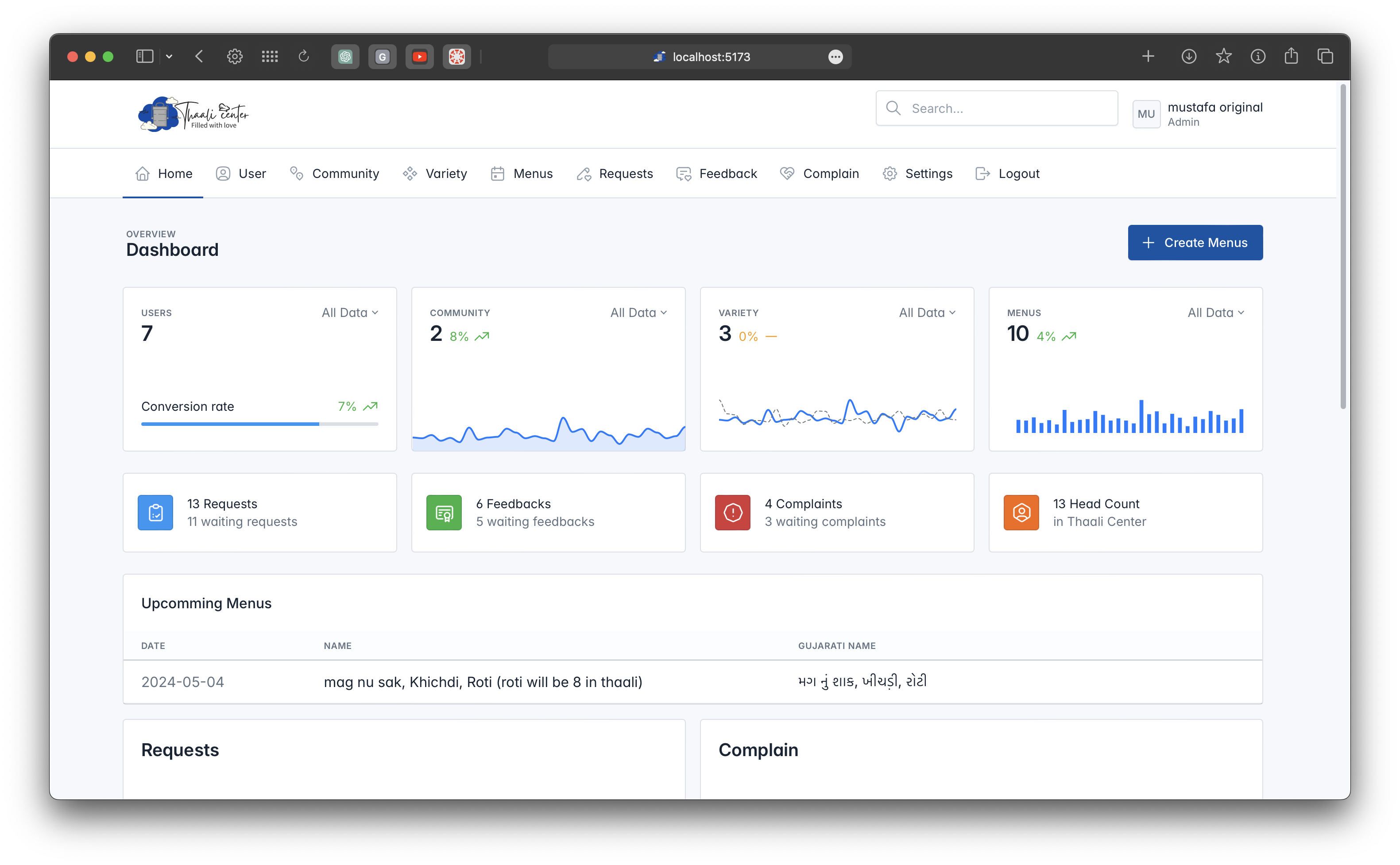
The Thaali Center project addresses the need for a comprehensive management system tailored to the unique requirements of businesses and communities. By leveraging modern technologies and best practices, the system aims to optimize workflow efficiency, promote user engagement, and facilitate effective customer feedback management and complaint resolution.

**Flowchart Diagram:**

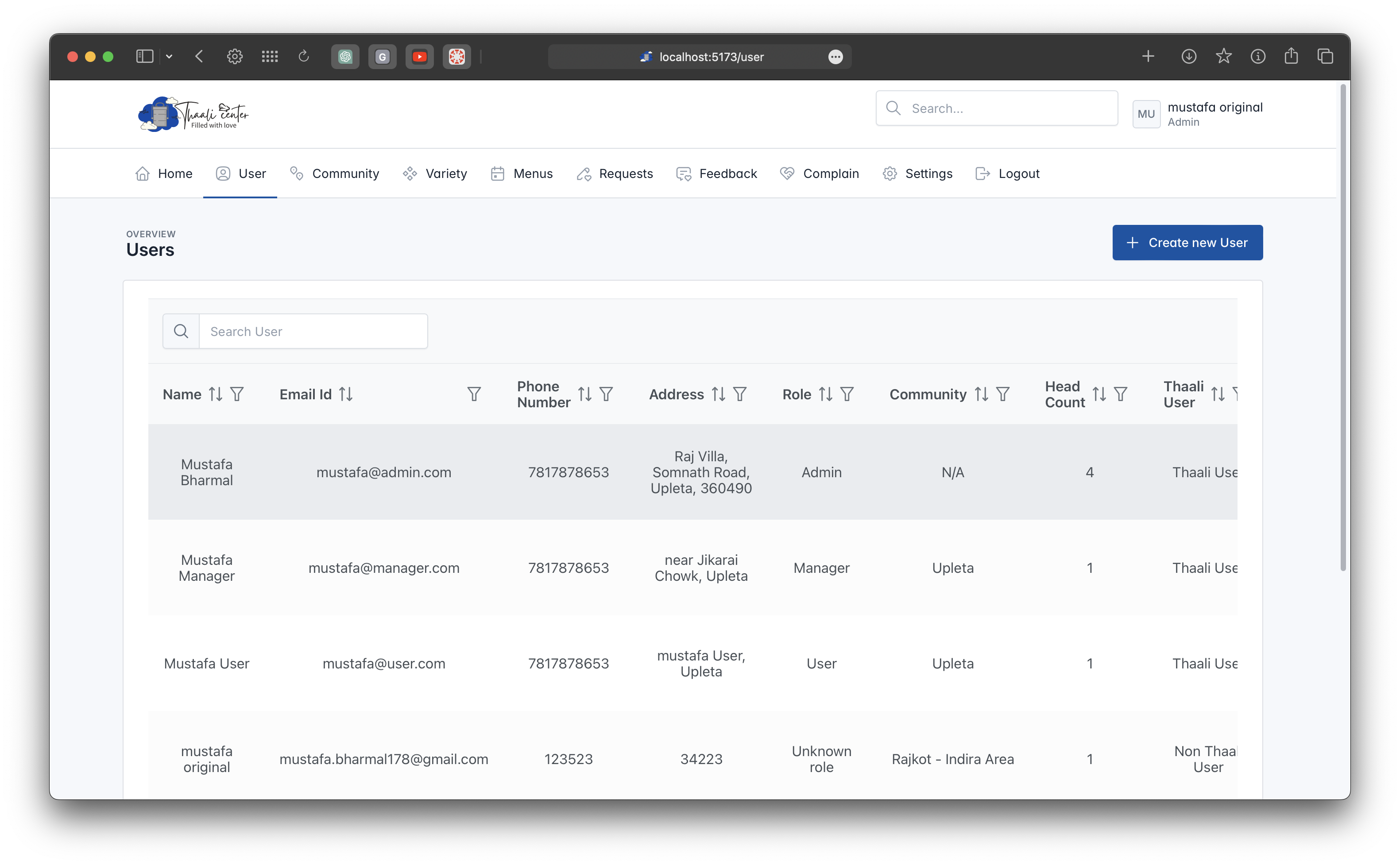


**Screenshots:**

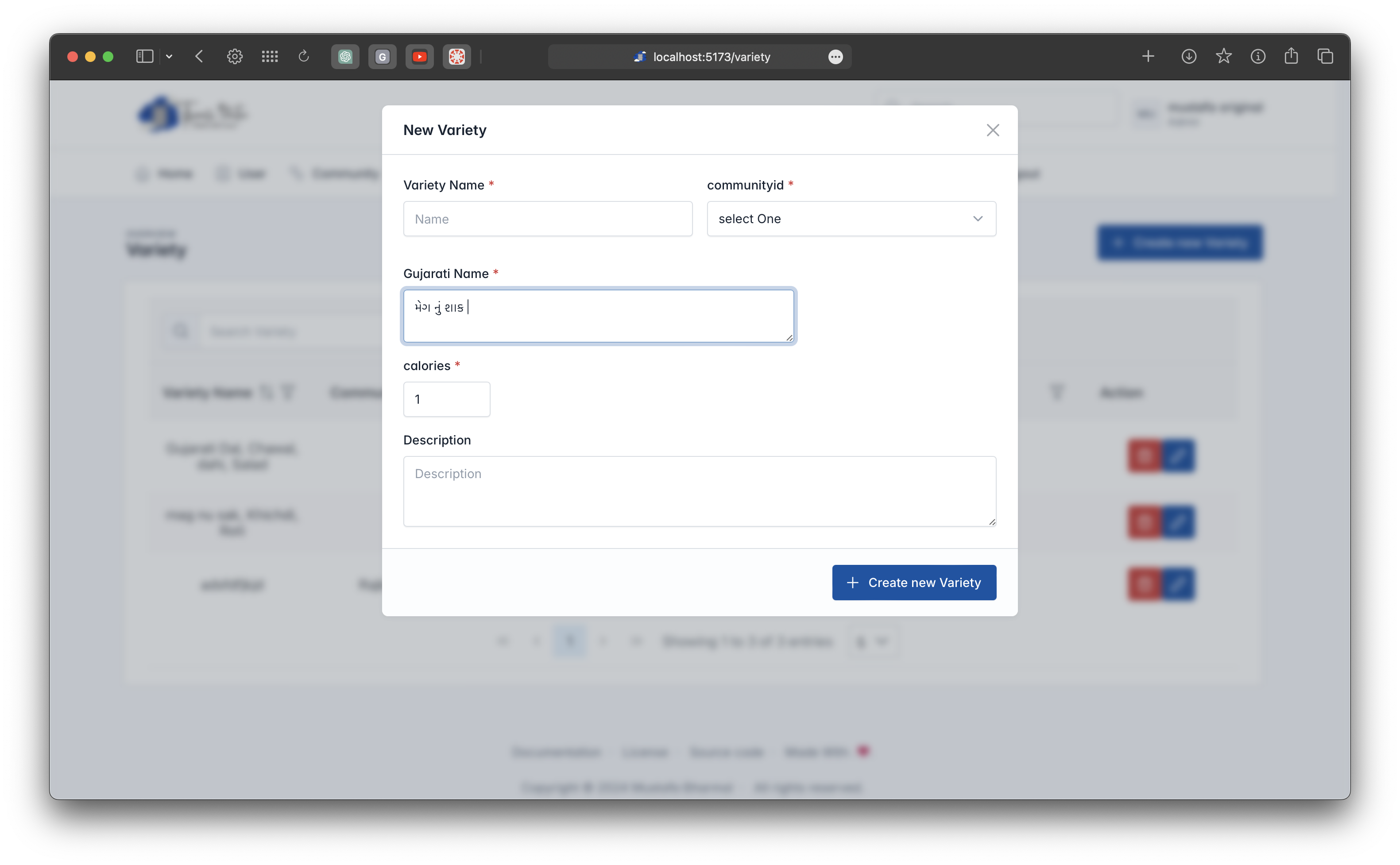
Dashboard for Admin/ Manger/ Users:



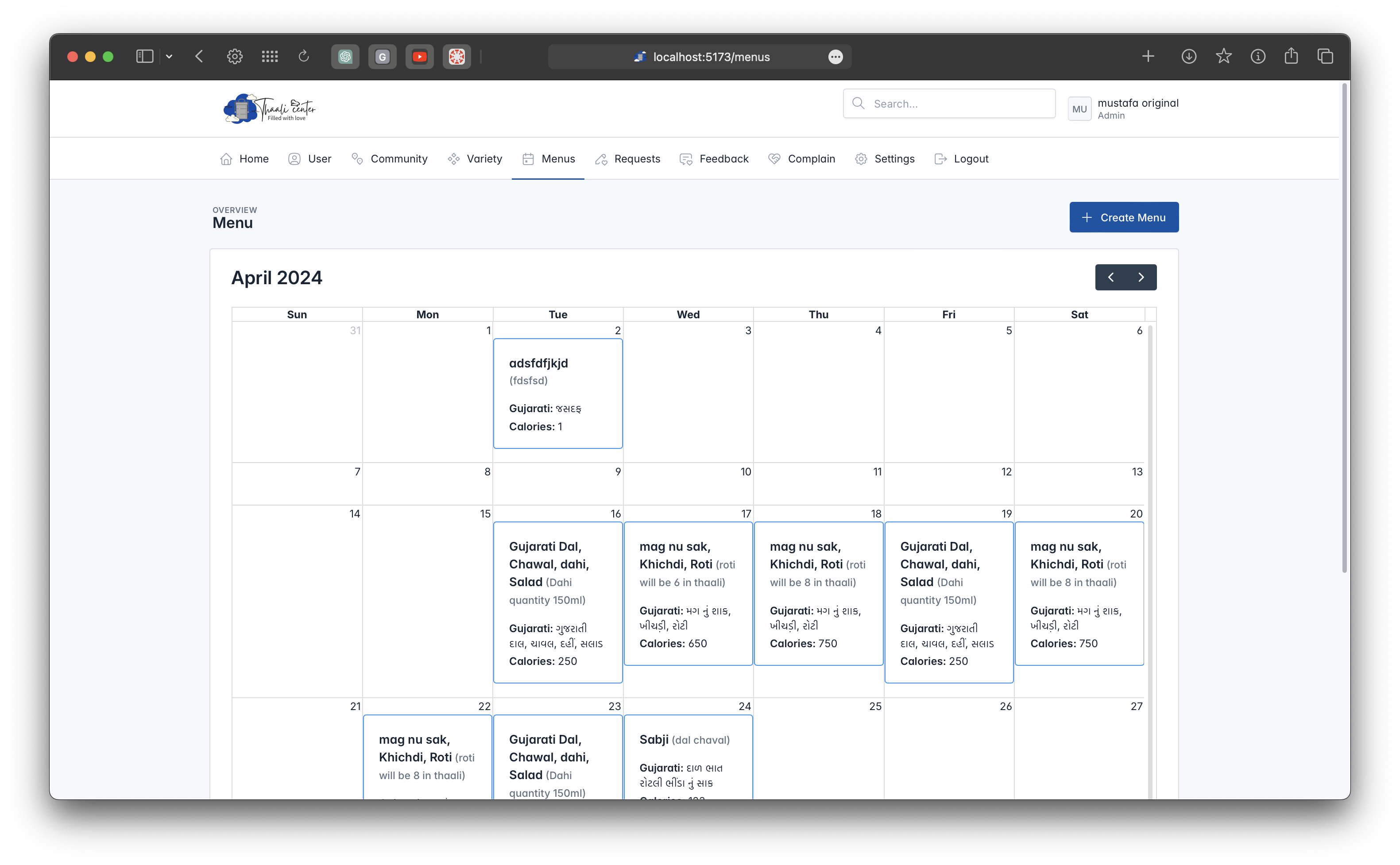
User Management for Admin/ Manger:



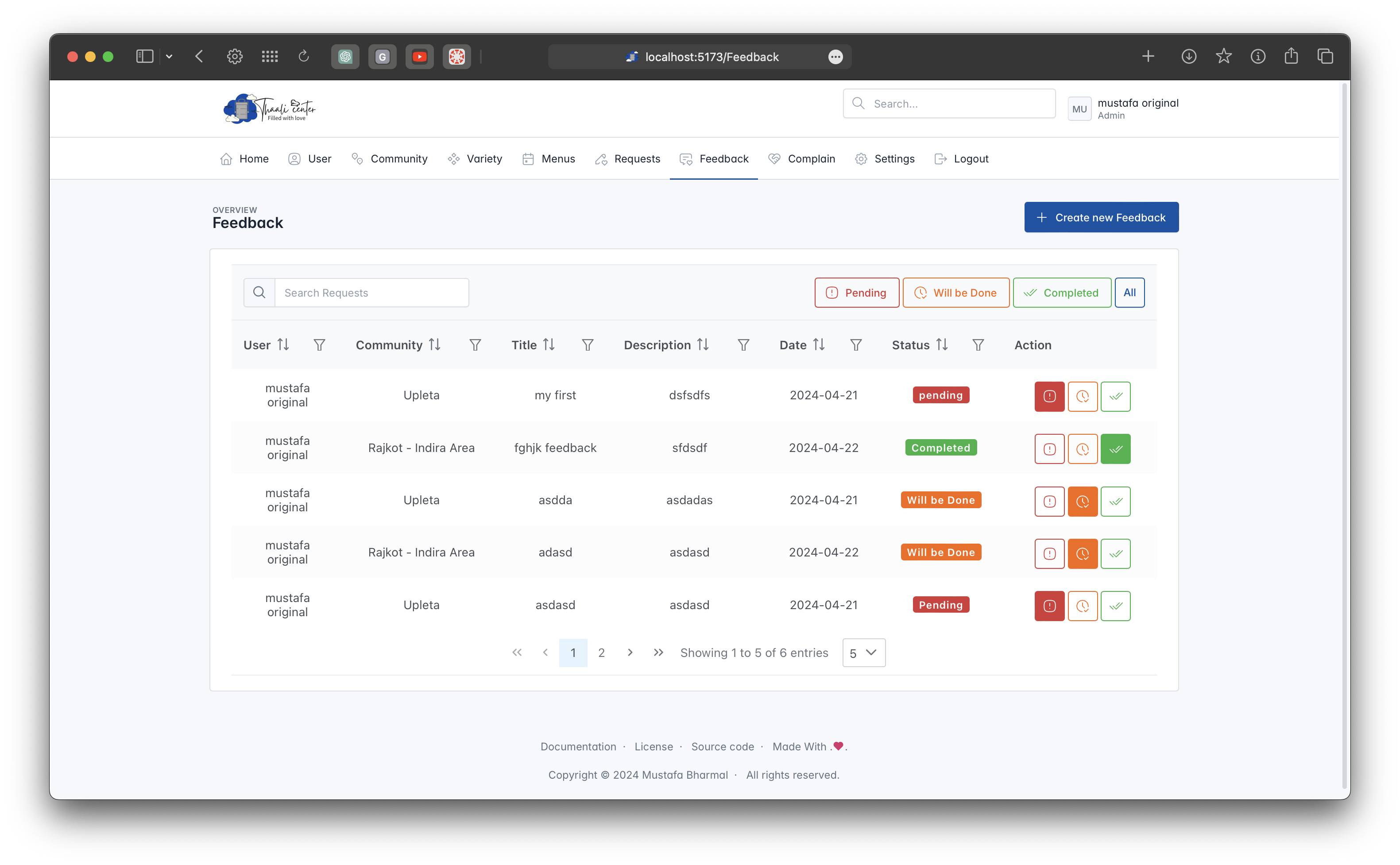
Variety Management for Admin/ Manger:



Menu Management for Admin/ Manger/ User:



Feedback Management for Admin/ Manger/ User:



**Code:**

Varirty Module for Admin/ Manger with Gujarati Translation:

const authCtx = useContext(AuthContext);

const isAdmin = authCtx.role === 0 || authCtx.role === "0";

const isManager = authCtx.role === 1 || authCtx.role === "1";

const isUser = authCtx.role === 2 || authCtx.role === "2";

const [formData, setFormData] = useState({

name: "",

calories: 1,

description: "",

gujaratiName: "",

communityid: isManager ? authCtx.communityid : "0",

status: 1,

createdat: Date.now(),

updatedat: Date.now(),

});

const [ComValues, setComValues] = useState([]);

// let input3 = formData.gujaratiName;

// enableTransliteration(input3, "gu");

// useEffect(() => {

// fetchComData();

// enableTransliteration(input3, "gu");

// }, []);

useEffect(() => {

isAdmin && (fetchComData());

const input = document.getElementById('data');

enableTransliteration(input, 'gu');

// console.log(input.value)

return () => {

// Clean up the transliteration when the component unmounts

// input.transliterator.disable();

disableTransliteration(input);

};

}, []);

const handleKeyUp = (e) => {

setFormData((prevData) => ({

...prevData,

gujaratiName: e.target.value,

}));

console.log(formData.gujaratiName);

};

const fetchComData = async () => {

try {

const response = await axios.get('http://localhost:3000/community', {

headers: {

authorization: `Mustafa ${authCtx.token}`,

},

withCredentials: true,

});

const transformedData = response.data.map(item => ({

\_id: item.\_id,

name: item.name,

address: `${item.address}`,

status: item.status,

createdat: item.createdat,

updatedat: item.updatedat,

}));

setComValues(transformedData);

} catch (error) {

console.error('Error fetching data:', error);

}

};

const handleChange = (e) => {

const { name, value } = e.target;

setFormData((prevData) => {

// Convert strings to integers for specific fields

const intValue = [

].includes(name)

? parseInt(value, 10)

: value;

// Format date fields

const formattedValue = name.endsWith("at")

? new Date(value).toISOString()

: intValue;

console.log(formattedValue);

return {

...prevData,

gujaratiName: document.getElementById('data').value,

[name]: formattedValue,

};

});

};

const handleSubmit = async (e) => {

e.preventDefault();

// console.log(formData);

try {

setFormData((prevData) => ({

...prevData,

gujaratiName: document.getElementById('data').value,

createdat: new Date().toISOString(),

updatedat: new Date().toISOString(),

}));

// console.log(formData);

const response = await fetch("http://localhost:3000/variety/add", {

method: "POST",

headers: {

authorization: `Mustafa ${authCtx.token}`,

"Content-Type": "application/json",

},

body: JSON.stringify(formData),

});

if (response.ok) {

console.log("User created successfully");

window.location.reload();

} else {

console.error("Failed to create user");

}

} catch (error) {

console.error("Error:", error);

}

};

**Thaali Center: Future Enhancements**

The Thaali Center is designed to evolve and adapt to the changing needs of users and communities. Future enhancements aim to enrich functionality, improve user experience, and ensure scalability. Here are potential future upgrades for the Thaali Center:

Integration of Analytics Dashboard

* Implement a comprehensive analytics dashboard to visualize user engagement, community trends, and menu preferences.
* Leverage data insights for informed decision-making and strategic planning.

Enhanced Multilingual Support

* Extend language support to include additional regional languages, catering to diverse user preferences.
* Enable seamless interaction with menus and system interfaces in multiple languages.

Automated Feedback Analysis

* Develop automated sentiment analysis tools to process user feedback and identify actionable insights.

Real-time Community Interaction Features

* Introduce real-time communication features for community members to collaborate, share updates, and organize events.
* Foster community engagement through interactive features and notifications.

Mobile Application Development

* Develop a dedicated mobile application for convenient access to Thaali Center features on smartphones and tablets.
* Optimize user experience with native app functionalities and offline capabilities.

Enhanced Security Measures

* Implement advanced security protocols, such as multi-factor authentication and encryption, to safeguard user data and system integrity.

Integration with IoT Devices

* Explore IoT integration for smart kitchen solutions, enabling automated menu updates and inventory management.

These future enhancements aim to elevate the Thaali Center into a robust, feature-rich platform that empowers users, fosters community engagement, and delivers unparalleled value to stakeholders. Each upgrade aligns with the project's vision of innovation and excellence in management solutions.

**References:**

References for the Thaali Center project encompass a range of sources that have contributed to the development, implementation, and enhancement of the system. These references include:

React Documentation

* Official documentation for React.js, used for building the frontend interface of the Thaali Center.
* Website: React Documentation

Express.js Documentation

* Official documentation for Express.js, utilized for developing the backend server and APIs.
* Website: Express.js Documentation

MongoDB Documentation

* MongoDB documentation providing guidance on schema design, queries, and database operations.
* Website: MongoDB Documentation

PrimeReact Documentation

* Documentation for PrimeReact, offering UI components and tools used to create the Thaali Center's frontend.
* Website: PrimeReact Documentation

Bootstrap Documentation

* Bootstrap documentation for designing responsive and attractive user interfaces.
* Website: Bootstrap Documentation

Java Spring Boot Documentation

* Documentation for Java Spring Boot, utilized for developing the feedback and complaint modules.
* Website: Spring Boot Documentation

Software Engineering Best Practices

* Various resources and textbooks on software engineering principles, design patterns, and best practices followed during development.
* Example: "Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin

These references serve as valuable sources of information, guidance, and inspiration throughout the lifecycle of the Thaali Center project, ensuring adherence to industry standards, efficiency in development, and continuous improvement of the system.