**Infosys Springboard 5.0 Internship**

**Project Report:**

**DebateHub: A Platform for Nuanced Discussions**

**Submitted By:**

**Keerthi Nune**

**Leela Lakshmi Karasa**

**Lalit Kishor**

**Table of Contents**

1. **Overview**1.1 Project Description  
   1.2 Objectives
2. **Features**2.1 User Features  
   2.2 Administrative Features
3. **Technology Stack**3.1 Frontend Technologies  
   3.2 Backend Technologies  
   3.3 Additional Libraries and Tools
4. **Modules**4.1 User Management  
   4.2 Debate and Options Management  
   4.3 Voting and Social Features
5. **Installation and Setup**5.1 Prerequisites  
   5.2 Steps to Run Locally
6. **API Documentation**6.1 Authentication Endpoints  
   6.2 Debate Management Endpoints  
   6.3 Voting and Social Features Endpoints
7. **Screenshots**7.1 User Dashboard  
   7.2 Admin Dashboard  
   7.3 Debate Details Page  
   7.4 Search and Filter Feature
8. **Conclusion**

**1.Overview**

**1.1 Project Statement:**

**DebateHub** is a web application designed to facilitate nuanced discussions and debates on a variety of topics. Users can create debate topics by posting questions with 2 to 7 options (viewpoints), allowing the community to participate actively. Each debate topic can be liked or disliked, providing a measure of community feedback. All debates are searchable for easy navigation.

To encourage meaningful engagement, each user is allocated 10 votes, which they can freely distribute across the available options. The voting outcomes are visually represented in graphical formats, offering a clear and intuitive understanding of the community's preferences.

**1.2 Objectives:**

The primary goal of DebateHub is to create a dynamic space where users can discuss, share ideas, and engage in democratic voting. Key objectives include:

* Providing a secure platform for creating and managing debates.
* Offering real-time vote tracking and graphical insights.
* Enabling easy moderation by administrators for content quality.
* Delivering an intuitive user experience with responsive design and seamless navigation.

**2. Features**

**2.1 User Features**

Users can register securely, create debates with multiple options, and participate in active discussions. They can cast votes, like/dislike topics, and view real-time graphical vote distributions. Advanced search and filtering tools enhance navigation, enabling users to explore topics of interest effortlessly.

**2.2 Administrative Features**

Admins have elevated privileges, such as the ability to moderate debates and manage user accounts. They can close debates to prevent further voting, review flagged content, and maintain the platform's integrity. Administrative tools are designed to ensure a balanced and respectful debating environment.

**3. Technology Stack**

**3.1 Frontend Technologies**

The frontend is developed using **React.js** for its component-based architecture and dynamic rendering. **Tailwind CSS** provides a clean and responsive design, while **React-Icons** enhances the UI with modern, intuitive icons. **React-Router-Dom** enables efficient client-side routing.

**3.2 Backend Technologies**

The backend leverages **Node.js** and **Express.js** to build RESTful APIs that support authentication, debate management, and voting functionalities. Data is efficiently handled through MongoDB, ensuring scalability and performance.

**3.3 Additional Libraries and Tools**

**Nodemailer** is used for email verification, ensuring secure account setup. **Socket.IO** powers real-time updates, while **Nodemon** facilitates automatic server refreshes during development.

**4. Modules**

**4.1 User Management**

* **Registration:** Users sign up with email and password. Passwords are hashed for security.
* **Email Verification:** A verification email is sent to confirm the user's email address.
* **Login:** Users log in with email and password, receiving a JWT token for session management.
* **Role-Based Redirection:** Users are redirected to dashboards based on their roles (e.g., Admin, Voter).

**4.2 Debate and Options Management**

* **Debate Creation:** Users create debates with 2-7 options, including title, description, and optional media.
* **Search & Filter:** Users can search and filter debates by category, popularity, or recency.
* **Statistics:** Graphs display vote distribution, and detailed statistics show participation and voting patterns.

**4.3 Voting and Liking**

* **Voting:** Users have 10 votes per debate, which they can distribute across options.
* **Real-Time Updates:** Vote counts update live, and users are notified when voting is complete.
* **Liking:** Users can like/dislike debates.

**5. Installation and Setup**

**5.1 Prerequisites**

Ensure that **Node.js**, **MongoDB**, and **Git** are installed. A modern browser is recommended for the best user experience. Obtain API keys for email and media functionalities if required.

1. **Install nodemon (globally)**:  
   Nodemon is a tool that automatically restarts the server whenever file changes are detected.

“npm install -g nodemon”

1. **Install nodemailer (locally)**:  
   Nodemailer is used for email functionalities such as verification.

“npm install nodemailer”

1. **Install other backend dependencies (if not already installed)**:  
   Ensure required backend libraries are installed, including Express and MongoDB.

“npm install express mongoose dotenv”

* 1. **Install react-router-dom**:  
     For handling routing in the React frontend.

“npm install react-router-dom”

* 1. **Install react-icons**:  
     For using Font Awesome and other icon libraries in the frontend.

“npm install react-icons”

* 1. **Install vite dependencies (if not already installed)**:  
     Ensure the project is initialized with Vite and required dependencies are installed.

“npm install”

**5.2 Steps to Run Locally**

Clone the repository, install dependencies for both frontend and backend, and configure environment variables. Start the frontend using **npm run dev** and the backend using **npm start**. Access the application at http://localhost:5173.

**6. API Documentation**

**6.1 Authentication Endpoints**

* POST /register: Registers a new user.
* POST /login: Authenticates user credentials and generates a JWT.
* POST /verify-email: Sends an email for the verification during register.
* POST /request-password-reset: Sends an email for password reset.

**6.2 Debate Management Endpoints**

* POST /debates: Creates a new debate with options.
* GET /alldebates: Fetches all debates with filtering options.
* Get /allusers: For fetching all the registered users.

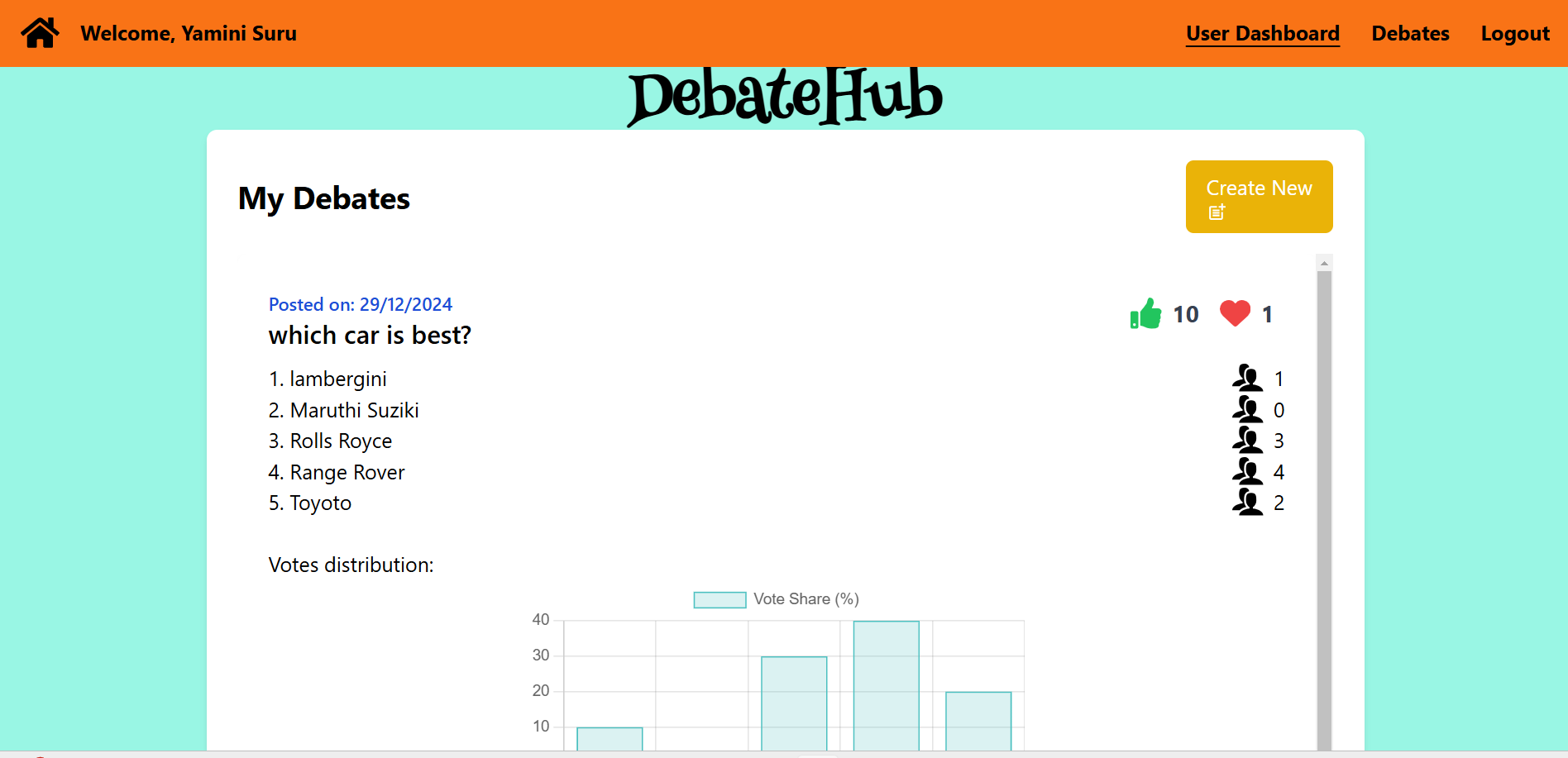
**6.3 Voting and Social Features Endpoints**

* POST /vote/:id: Records a user’s votes on a debate.
* POST /like/:id and /dislike/:id: Likes/dislikes a debate.

**7. Screenshots**

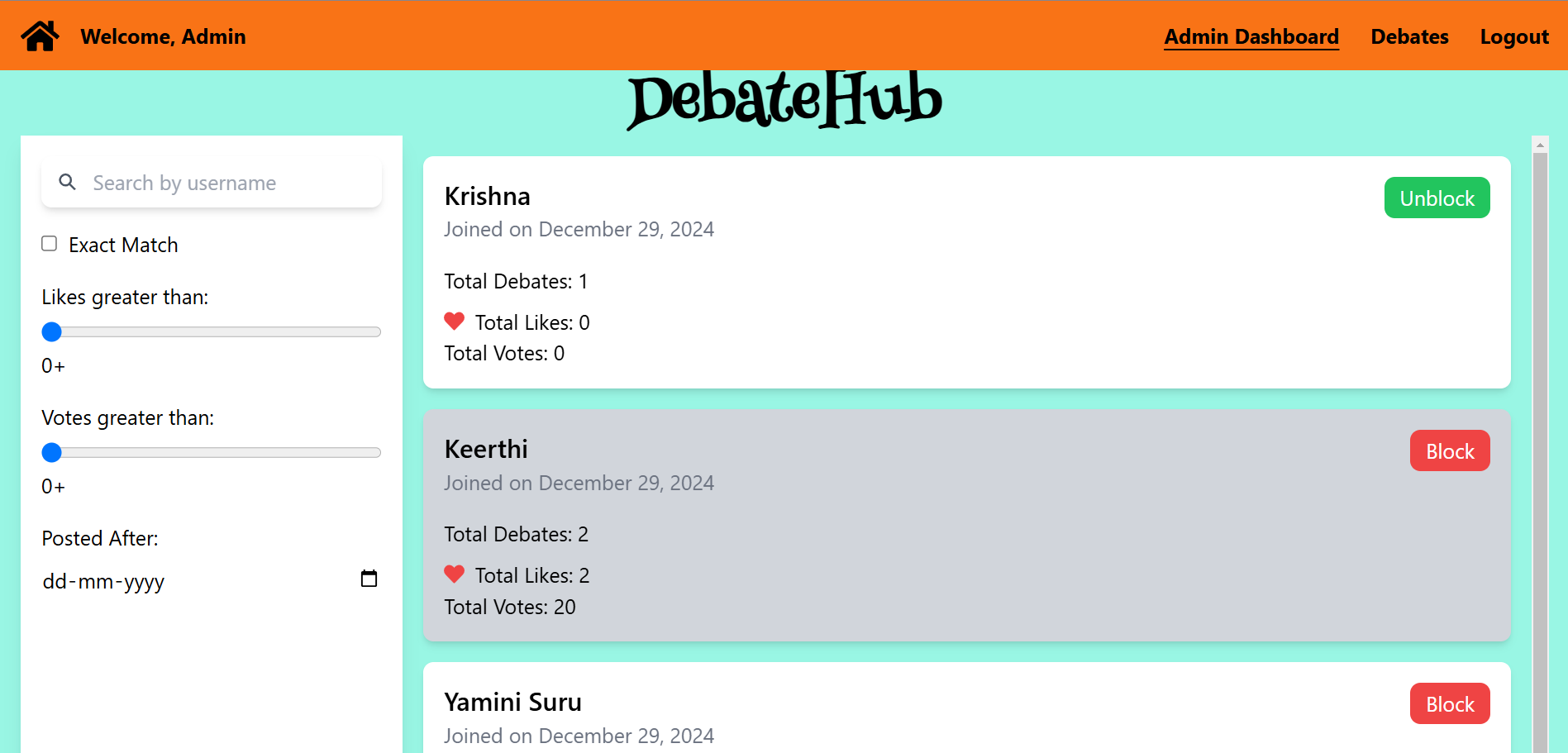
**7.1 User Dashboard**

The user dashboard displays a list of active debates and blocked debates with options , no.of likes and vote.The blocked debates will be displayed in red colour and wont be accessible to the creator of that debate until it is unblocked.

**

**7.2 Admin Dashboard**

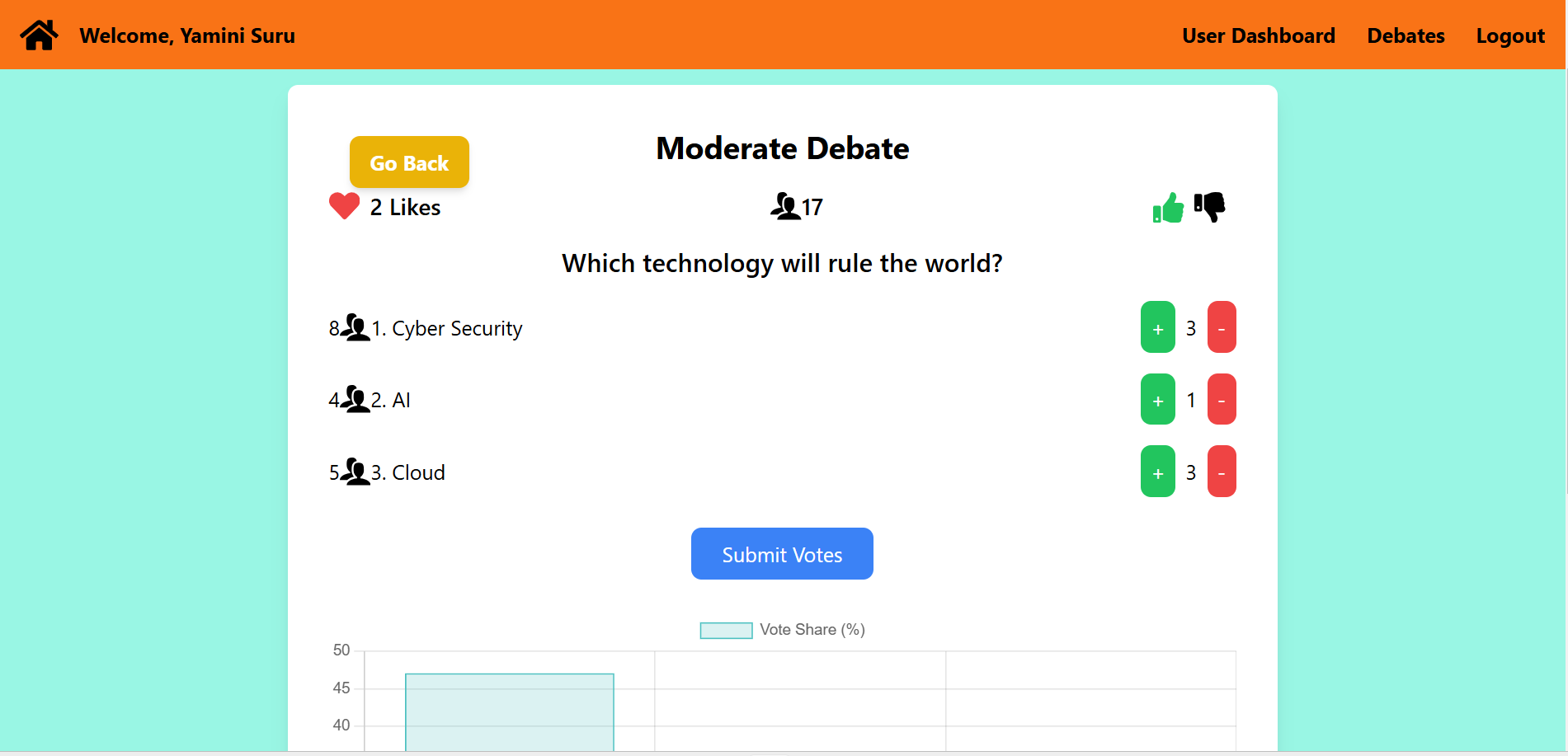
The admin dashboard provides tools for moderating debates and managing users.

**

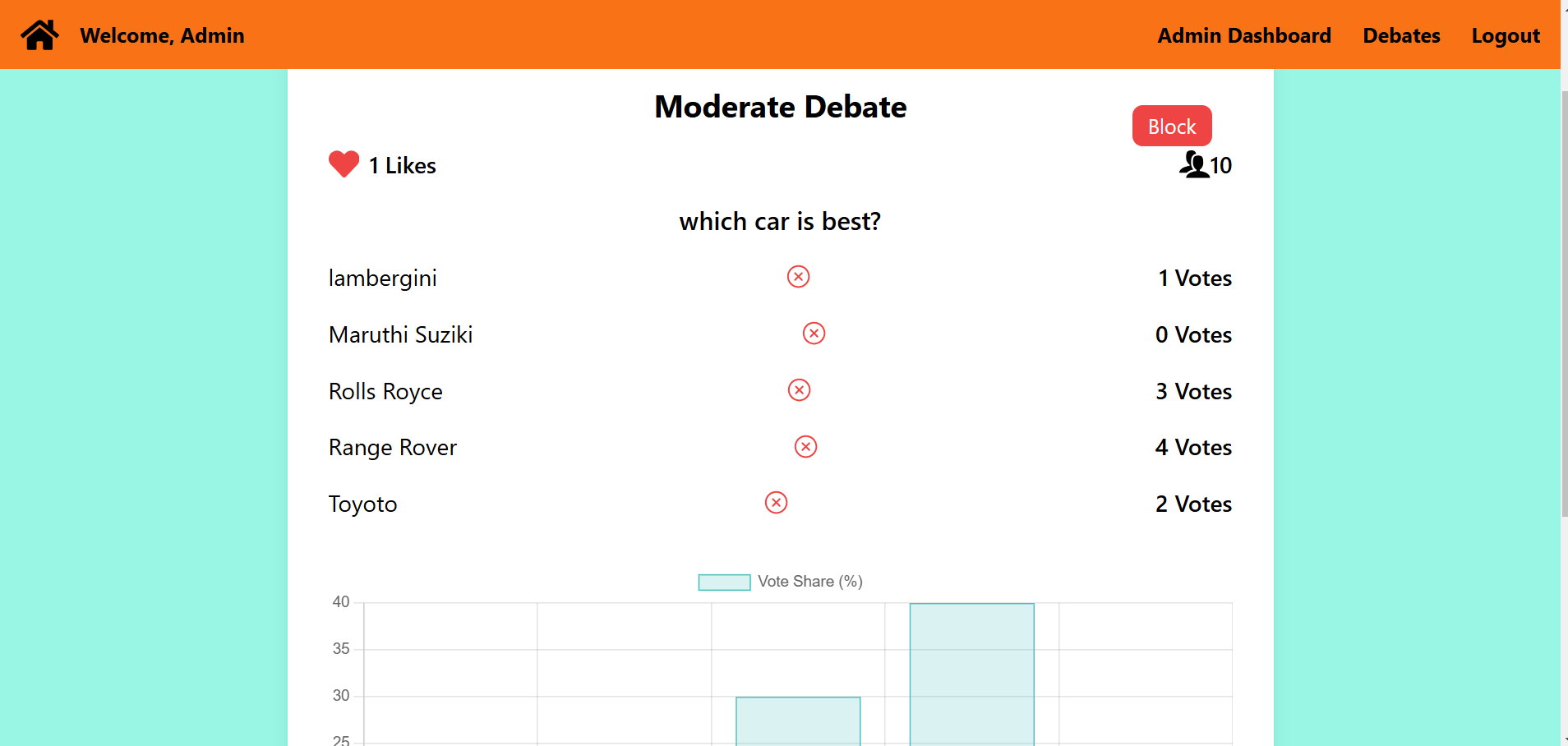
**7.3 Debate Details Page**

Shows the debate question, options, vote distribution, and likes.

*Moderate Debates on Userside.*

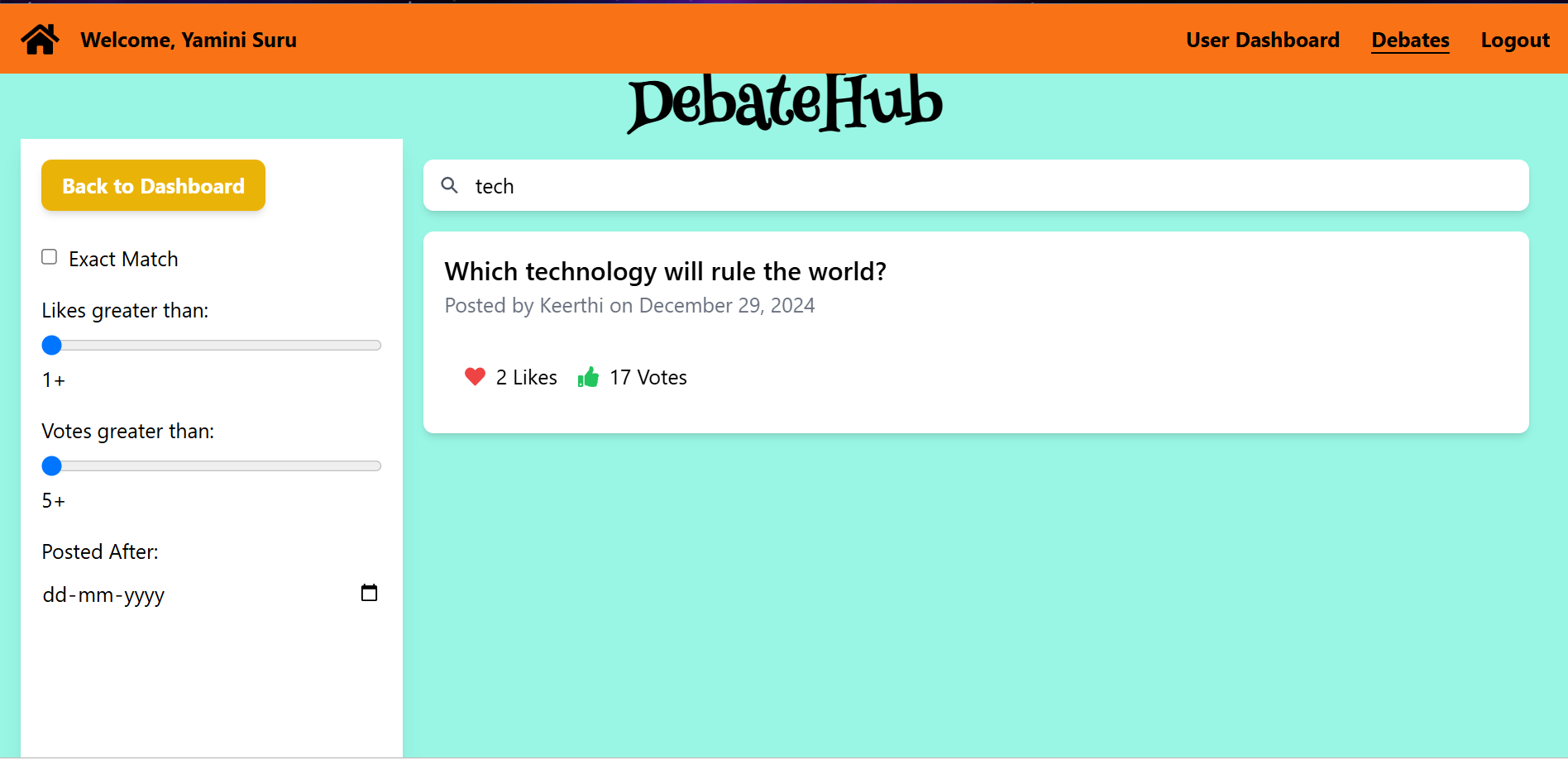
**

*Moderate Debates on Admin side.*

**

**7.4 Search and Filter Feature**

An intuitive interface for exploring debates by keywords, categories, or popularity.



**8. Conclusion**

**DebateHub** is a user-friendly platform designed to promote meaningful discussions. With its easy navigation, real-time voting, and social features, users can engage actively in debates. The secure login and role-based access ensure a safe and personalized experience for all users.

The platform’s interactive features, such as vote tracking, liking, and commenting, create an engaging environment for sharing ideas. DebateHub is a valuable tool for anyone looking to participate in thoughtful and democratic debates.