Real-Time Polling Application

A full-stack real-time polling application built with React, Node.js, Express, and WebSocket support.

Features

- Real-time poll creation and voting
- User authentication with JWT
- Live vote updates via WebSockets
- Responsive design with Tailwind CSS
- Secure API with rate limiting and validation

Prerequisites

Make sure you have the following installed on your system:

- Node.js (version 18.0.0 or higher)
- npm (comes with Node.js)
- Git

Installation

1. Clone the Repository

```bash
git clone https://github.com/SwAsTiK-KuL/realtime_voting.git
```

2. Navigate to Project Directory

```bash
cd realtime_voting
```

3. Install Backend Dependencies

```bash
cd backend
npm install
cd ..
```

4. Install Frontend Dependencies

```bash
npm install
```

5. Environment Setup

Create a `.env` file in the `backend` directory:

```bash
cd backend
# Create .env file with your configuration
# Example:
# DATABASE_URL="your-database-connection-string"
# JWT_SECRET="your-jwt-secret-key"
# PORT=3001
```

6. Database Setup (if using Prisma)

```bash
# Still in backend directory
npm run db:generate
npm run db:push
# Optional: Seed database with sample data
npm run db:seed
```

7. Start the Application

Return to the root directory and start both frontend and backend:

```bash
cd ..
npm run dev
```

This command will start:

- Backend server on `http://localhost:3001`
- Frontend development server on `http://localhost:5173`

Available Scripts

Root Directory

- `npm run dev` - Start both frontend and backend concurrently
- `npm run frontend:dev` - Start only the frontend development server
- `npm run backend:dev` - Start only the backend development server
- `npm run build` - Build the frontend for production
- `npm run install:all` - Install dependencies for both frontend and backend

Backend Directory

- `npm start` - Start the production server
- `npm run dev` - Start the development server with nodemon
- `npm run db:generate` - Generate Prisma client
- `npm run db:push` - Push database schema changes
- `npm run db:migrate` - Run database migrations
- `npm run db:seed` - Seed the database with sample data
- `npm run db:studio` - Open Prisma Studio

Project Structure

```
realtime_voting/
├── backend/              # Backend API server
│   ├── prisma/           # Database schema and migrations
│   ├── server.js         # Main server file
│   └── package.json      # Backend dependencies
├── src/                  # Frontend React application
│   ├── components/       # React components
│   ├── App.jsx           # Main app component
│   ├── main.jsx          # React entry point
│   └── index.css         # Tailwind CSS styles
├── public/               # Static assets
├── package.json          # Frontend dependencies and scripts
├── vite.config.js        # Vite configuration
├── tailwind.config.js    # Tailwind CSS configuration
└── README.md             # This file
```

Usage

1. Open your browser and navigate to `http://localhost:5173`
2. Register a new account or use the sample accounts provided on the login page

3. Create polls and share them with others
4. Vote on polls and see results update in real-time
5. View your dashboard to manage your polls

Technology Stack

Frontend

- React 19
- Vite (build tool)
- Tailwind CSS
- Socket.IO Client
- Lucide React (icons)

Backend

- Node.js
- Express.js
- Socket.IO (WebSockets)
- Prisma ORM
- JWT Authentication
- bcryptjs (password hashing)
- Joi (validation)

Contributing

1. Fork the repository
2. Create a feature branch (`git checkout -b feature/amazing-feature`)
3. Commit your changes (`git commit -m 'Add some amazing feature'`)
4. Push to the branch (`git push origin feature/amazing-feature`)
5. Open a Pull Request

License

This project is licensed under the MIT License - see the LICENSE file for details.

Author

Swastik Kulkarni

Support

If you encounter any issues during installation or usage, please check:

1. Node.js version compatibility
2. All dependencies are installed correctly
3. Environment variables are set properly
4. Database connection is working

For additional help, please open an issue on the GitHub repository.