



SocialBoard - Image Sharing Platform

A modern and responsive social media platform designed for image sharing and connecting with others. Built using React, Node.js, and MongoDB, SocialBoard provides a seamless and engaging experience for users and admins alike.



Live Project Link

[Click here to view the live project](#)

✦ Features

- **User Authentication & Authorization**
 - **Image Upload and Sharing**
 - **User Profiles & Social Handles**
 - **Customizable Themes** (Light & Green)
 - **Admin Dashboard** for User Management
 - **Responsive Design** for all Devices
 - **Real-time Updates**
 - **Role-Based Access Control**
-

🔧 Tech Stack

Frontend

- **React**
- **Tailwind CSS**
- **Zustand** (State Management)
- **React Router**
- **React Hook Form**
- **Zod** (Form Validation)
- **TypeScript**
- **Vite**

Backend

- **Node.js**
 - **Express.js**
 - **MongoDB**
 - **JWT Authentication**
 - **Multer** (File Upload)
 - **bcrypt.js**
-



Installation

1. Clone the repository:

```
git clone https://github.com/shibbu04/social-board.git
cd social-board
```

2. Install dependencies for both frontend and backend:

```
# Install frontend dependencies
cd frontend
npm install

# Install backend dependencies
cd backend
npm install
```

3. Set up environment variables:

```
# In backend folder, create .env file
PORT=5000
MONGODB_URI=your_mongodb_uri
JWT_SECRET=your_jwt_secret
ADMIN_EMAIL=admin@admin.com
ADMIN_PASSWORD=admin123
```

Running the Project

1. Start the backend server:

```
# In backend folder
npm start
```

2. Start the frontend development server:

```
# In root folder
npm run dev
```

Project Structure

```
socialboard/
├── frontend/
│   ├── components/    # Reusable UI components
│   ├── pages/         # Page components
│   ├── store/         # Zustand store configurations
│   ├── lib/           # Utility functions and configurations
│   └── types/         # TypeScript type definitions
├── backend/
│   ├── src/
│   │   ├── controllers/ # Route controllers
│   │   ├── models/      # MongoDB models
│   │   ├── routes/      # API routes
│   │   ├── middleware/  # Custom middleware
│   │   └── utils/       # Utility functions
│   └── uploads/        # Uploaded images storage
└── public/            # Static assets
```

Authentication

- **Register** with email and password
- **Login** with credentials
- **JWT-Based Authentication**
- **Protected Routes**
- **Role-Based Access** (User, Admin, Superadmin)

User Roles

- **User**: Can create, edit, and delete their own posts
- **Admin**: Can manage users and all posts
- **Superadmin**: Has full system access

Themes

- Light theme (Default)
- Green theme
- Persistent theme selection

Temporary Logins

For testing purposes, you can use the following temporary credentials to log in as a user or admin:

Temporary User Login

- **Username**: `user@user.com`
- **Password**: `user123`

Temporary Admin Login

- **Username:** admin@admin.com
- **Password:** admin123

These credentials are valid for a limited time and are meant for demonstration purposes.



Developed By

Shivam Singh



Connect with me

Portfolio



GitHub

LinkedIn

Developed with by Shivam Singh