Dashboard Login System with Hierarchy

A modern, professional dashboard system with hierarchical user management built with Node.js, Express, MySQL, and featuring glassmorphism design with smooth animations.

Features

- 🔐 Secure login system with session management
- 👥 Hierarchical user roles (Admin, Manager, User)
- ✨ Interactive glassmorphism UI design
- 📊 Real-time dashboard statistics
- 🎨 Smooth animations and transitions
- 📱 Fully responsive design
- 🔒 Role-based access control
- 👤 User CRUD operations for admins/managers
- 📝 Activity logging system

Tech Stack

- **Backend**: Node.js, Express.js
- **Database**: MySQL
- **Session**: express-session
- **Security**: bcryptjs for password hashing
- **Template Engine**: EJS
- **Styling**: Custom CSS with glassmorphism effects

Project Structure

```
dashboard-login-system/
├── views/
│   ├── partials/
│   │   ├── header.ejs
│   │   └── sidebar.ejs
│   ├── login.ejs
│   ├── dashboard.ejs
│   └── users.ejs
├── public/
│   ├── css/
│   │   └── style.css
│   └── js/
│       ├── login.js
│       ├── dashboard.js
│       └── users.js
├── server.js
├── database.sql
├── package.json
├── .env
├── .gitignore
└── README.md
```

Installation & Setup

1. Clone the Repository

```
bash

git clone <your-repo-url>
cd dashboard-login-system
```

2. Install Dependencies

```
bash

npm install
```

3. Database Setup

Create a MySQL database and run the SQL script:

```
bash

mysql -u root -p < database.sql
```

Or manually execute the `database.sql` file in your MySQL client.

4. Environment Configuration

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

```
env
```

```
# Database Configuration
DB_HOST=your_railway_mysql_host
DB_USER=root
DB_PASSWORD=your_mysql_password
DB_NAME=dashboard_db
DB_PORT=3306

# Server Configuration
PORT=3000
SESSION_SECRET=your_super_secret_key_change_this

# Admin Default Credentials
ADMIN_EMAIL=admin@dashboard.com
ADMIN_PASSWORD=Admin@123
```

5. Run the Application

Development:

```
bash

npm run dev
```

Production:

```
bash

npm start
```

The application will run on `http://localhost:3000`

Deployment to Railway.app

Step 1: Create Railway Account

1. Go to Railway.app
2. Sign up with GitHub

Step 2: Create MySQL Database

1. Click "New Project"
2. Select "Provision MySQL"
3. Copy the connection details

Step 3: Deploy Application

1. Click "New" → "GitHub Repo"
2. Select your repository
3. Add environment variables in Railway dashboard:
   - `DB_HOST` - From MySQL service
   - `DB_USER` - From MySQL service
   - `DB_PASSWORD` - From MySQL service
   - `DB_NAME` - From MySQL service
   - `DB_PORT` - From MySQL service (usually 3306)
   - `SESSION_SECRET` - Generate a secure random string
   - `NODE_ENV` - Set to `production`

Step 4: Initialize Database

Connect to Railway MySQL and run the `database.sql` script to create tables and default admin user.

Step 5: Deploy

Railway will automatically deploy your application. Access it via the generated Railway URL.

Default Login Credentials

- **Email**: admin@dashboard.com
- **Password**: Admin@123

⚠️ **IMPORTANT**: Change the default admin password after first login!

User Roles & Permissions

Admin

- Full access to all features
- Can create/edit/delete all users
- Can create other admins
- Access to all dashboard features

Manager

- Can create/edit/delete regular users

- Cannot manage admin accounts
- Access to dashboard and user management

User

- Basic dashboard access
- Cannot access user management
- View-only permissions

Features Breakdown

1. Login System

- Secure password hashing with bcryptjs
- Session-based authentication
- Activity logging
- Error handling with user feedback

2. Dashboard

- Real-time statistics
- User activity tracking
- Recent users list
- Animated statistics cards

3. User Management

- Create new users (Admin/Manager only)
- Edit user details
- Activate/Deactivate users
- Delete users
- Role assignment

4. Security Features

- Password hashing
- Session management
- Role-based access control

- SQL injection prevention
- XSS protection

5. UI/UX Features

- Glassmorphism design
- Smooth animations
- Responsive layout
- Loading states
- Toast notifications
- Modal dialogs

API Endpoints

Authentication

- `GET /login` - Login page
- `POST /login` - Login authentication
- `GET /logout` - Logout user

Dashboard

- `GET /dashboard` - Dashboard page
- `GET /users` - User management page

User Management (Admin/Manager only)

- `POST /users/create` - Create new user
- `POST /users/update` - Update user
- `POST /users/delete` - Delete user

Database Schema

Users Table

```
sql
```

```
- id (INT, PRIMARY KEY)
- name (VARCHAR)
- email (VARCHAR, UNIQUE)
- password (VARCHAR)
- role (ENUM: admin, manager, user)
- status (ENUM: active, inactive)
- created_by (INT, FOREIGN KEY)
- created_at (TIMESTAMP)
- updated_at (TIMESTAMP)
```

Activity Logs Table

```sql
- id (INT, PRIMARY KEY)
- user_id (INT, FOREIGN KEY)
- action (VARCHAR)
- details (TEXT)
- ip_address (VARCHAR)
- created_at (TIMESTAMP)
```

Development

Adding New Features

1. Create new routes in `server.js`
2. Add views in `views/` directory
3. Add styles in `public/css/style.css`
4. Add JavaScript in `public/js/`

Running Tests

```bash
npm test
```

Troubleshooting

Database Connection Issues

- Verify MySQL is running
- Check `.env` credentials

- Ensure database exists

Session Issues

- Check `SESSION_SECRET` is set
- Clear browser cookies
- Restart the server

Railway Deployment Issues

- Verify all environment variables
- Check MySQL connection string
- Review Railway logs

Security Best Practices

1. **Change Default Credentials**: Always change admin password after setup
2. **Secure SESSION_SECRET**: Use a strong, random string
3. **HTTPS in Production**: Railway provides HTTPS by default
4. **Regular Updates**: Keep dependencies updated
5. **Environment Variables**: Never commit `.env` file

Contributing

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

License

MIT License - feel free to use this project for personal or commercial purposes.

Support

For issues and questions:

- Create an issue on GitHub

- Contact: your-email@example.com

Acknowledgments

- Design inspired by modern glassmorphism trends
- Icons from Lucide Icons
- Built with love using Node.js and Express

Happy Coding! 🚀