🌍 Story Swap - Hyperlocal Travel Stories Platform

> **A premium travel storytelling platform where users share location-based stories and discover hidden gems through authentic local experiences.**

📖 Project Overview

Story Swap is a sophisticated full-stack web application that enables travelers and locals to share location-specific stories, creating an authentic travel discovery platform. Users can upload stories tied to specific geographical locations, explore stories from others, and build a community around travel experiences.

🎯 Key Features

- 📍 **Location-Based Storytelling** - Stories are geotagged to specific locations
- 🔐 **User Authentication** - Secure login/registration with persistent sessions
- 📱 **Dynamic Island Navigation** - Premium iOS-inspired navigation with glassmorphism effects
- 👤 **Personalized User Experience** - User-specific story management and profiles
- 🗺️ **Interactive Story Discovery** - Explore stories by location and user
- 📝 **Rich Text Editor** - Advanced story creation with media support
- 🌟 **Modern UI/UX** - Travel-inspired design with smooth animations

🛠️ Technology Stack

Frontend

- **React 18** with TypeScript
- **Vite** for fast development and building
- **Tailwind CSS** for styling
- **Shadcn/UI** components
- **React Router** for navigation
- **TanStack Query** for data fetching
- **Custom Authentication Context**

Backend

- **Node.js** with Express.js
- **MongoDB** database with Mongoose ODM
- **JWT** authentication
- **CORS** configuration
- **RESTful API** architecture
- **Swagger** API documentation

Database

- **MongoDB** with collections for:

- Users (with authentication)
- Stories (location-based content)
- Comments and engagement data
- Geographic location data

Development Tools

- **TypeScript** for type safety
- **ESLint** for code quality
- **PostCSS** for CSS processing
- **Git** version control

🚀 Quick Start

Prerequisites

- Node.js (v18+ recommended)
- MongoDB (v7.0+ recommended)
- npm or yarn

One-Command Setup

```
./setup.sh
```

Start Application

```
# Start both frontend and backend
./start-all.sh

# Or start individually
./start-backend.sh    # Backend only
./start-frontend.sh   # Frontend only
```

🌟 Features

🔒 Security & Secrets (Important)

Recent cleanup removed hardcoded API / widget keys. To keep the project secure:

Area	Action
Frontend AgentX key	Provide via `frontend/vite-frontend/.env.local` as `VITE_AGENTX_KEY=your_key`
Backend JWT	Set `JWT_SECRET` and `REFRESH_TOKEN_SECRET` in `backend/.env` (never commit)

Area	Action
Map / external APIs	Store in env vars, not source files
Rotation	If a key was exposed, generate a new one → update env → redeploy → revoke old

Steps:

1. Copy `backend/.env.example` to `backend/.env` and fill values.
2. Copy `frontend/vite-frontend/.env.example` to `.env.local` and add keys.
3. Add any deployment secrets via your hosting provider's secret manager.
4. Never push real credentials – only `*.example` templates.

Fallback behavior: If `VITE_AGENTX_KEY` is absent, AI chat returns a safe placeholder message and logs a console warning.

To scrub an already committed secret (history rewrite example):

```
git filter-repo --invert-paths --path path/to/file-with-secret
git push --force origin main
```

> Treat any key you pasted into a public place as compromised—rotate it.

🗺️ Enhanced Map Integration

- **MapTiler Integration** with API key `NSO8JuqWXOqh8UZs5tpY`
- **Interactive Map Toggle** in Explore page
- **Story Location Markers** with clickable popups
- **Auto-centering** and smart zoom calculation
- **Professional Styling** with MapTiler Streets

🖼️ Enhanced UI/UX

- **Rotating Story Images** with error fallbacks
- **Enhanced Story Cards** with hover effects
- **Better Engagement Display** (likes, views, comments)
- **Professional Design** with gradients and animations
- **Responsive Layout** for all devices

🔐 Authentication

- **Demo Credentials** for easy testing
- **One-click Login** buttons
- **Role-based Access** (Admin/User)

📱 Pages & Features

- 🏠 **Home** - Hero section with call-to-action

- 🔍 **Explore** - Browse stories with interactive map
- 🗺️ **Map View** - Full-screen map experience
- 📝 **My Stories** - Story management dashboard
- ✏️ **Edit Story** - Rich story editing interface
- ➕ **Submit Story** - Create new stories
- 👤 **Profile** - User profile management

🏗️ Project Structure

```
story-swap-locale/
├── frontend/
│   └── vite-frontend/
│       ├── src/
│       │   ├── components/        # Reusable UI components
│       │   │   ├── Navigation.tsx # Dynamic Island navbar
│       │   │   ├── StoryCard.tsx  # Story display cards
│       │   │   └── ui/            # Shadcn/UI components
│       │   ├── pages/             # Route components
│       │   │   ├── Index.tsx      # Landing page
│       │   │   ├── Explore.tsx    # Story discovery
│       │   │   ├── MyStories.tsx  # User's stories
│       │   │   └── SubmitStory.tsx# Story creation
│       │   ├── contexts/          # React contexts
│       │   │   └── AuthContext.tsx# Authentication state
│       │   ├── lib/               # Utilities and API
│       │   └── types/             # TypeScript definitions
│       ├── package.json
│       └── vite.config.ts
├── backend/
│   ├── server.js                  # Express server entry
│   ├── controllers/               # Route handlers
│   │   ├── authController.js      # Authentication logic
│   │   ├── storyController.js     # Story CRUD operations
│   │   └── userController.js      # User management
│   ├── routes/                    # API routes
│   ├── middleware/                # Custom middleware
│   └── config/                    # Database & app config
├── database/
│   ├── models/                    # Mongoose schemas
│   │   ├── User.js               # User model
│   │   ├── Story.js              # Story model
│   │   └── Location.js           # Location model
│   └── seedData.js               # Sample data
└── docs/                          # Documentation
```

🔧 Configuration

Environment Variables

Backend (.env)

```
NODE_ENV=development
PORT=3001
MONGODB_URI=mongodb://localhost:27017/hyperlocal-story-swap
JWT_SECRET=your-jwt-secret
```

Frontend (.env)

```
VITE_MAPTILER_API_KEY=NSO8JuqWXOqh8UZs5tpY
VITE_API_URL=http://localhost:3001
VITE_APP_NAME=Story Swap
```

Database Setup

The application uses MongoDB. The setup script will:

1. Start MongoDB automatically (if using Homebrew)
2. Create the database on first connection
3. Seed with demo data using admin credentials

🎯 Demo Credentials

Admin Account:

- Email: `admin@example.com`
- Password: `test1234`

User Account:

- Email: `rita@example.com`
- Password: `test1234`

Additional User:

- Email: `sam@example.com`
- Password: `test1234`

🌐 Access URLs

After starting the application:

- **Frontend:** http://localhost:8080 or http://localhost:8081
- **Backend API:** http://localhost:3001
- **API Documentation:** http://localhost:3001/api/docs

📋 Manual Setup (Alternative)

If you prefer manual setup:

1. Install Dependencies

```
# Backend
cd backend && npm install

# Frontend
cd frontend/vite-frontend && npm install

# Database (if needed)
cd database && npm install
```

2. Start MongoDB

```
# Using Homebrew
brew services start mongodb-community@7.0

# Or manually
mongod --config /usr/local/etc/mongod.conf
```

3. Start Backend

```
cd backend && node server.js
```

4. Start Frontend

```
cd frontend/vite-frontend && npm run dev
```

🗺️ MapTiler Setup

The application comes pre-configured with MapTiler API key. To use your own:

1. Get API key from MapTiler Cloud
2. Update `VITE_MAPTILER_API_KEY` in `frontend/vite-frontend/.env`
3. Restart frontend server

See `docs/MAPTILER_SETUP.md` for detailed instructions.

🚫 Troubleshooting

Common Issues

MongoDB Connection Failed:

```
brew services start mongodb-community@7.0
```

Port Already in Use:

- Frontend will automatically try port 8081 if 8080 is busy
- For backend, stop other services using port 3001

Missing Dependencies:

```
./setup.sh  # Re-run setup script
```

Map Not Loading:

- Check console for MapTiler API errors
- Verify `VITE_MAPTILER_API_KEY` in `.env`
- Restart frontend after env changes

Reset Database

```
cd backend && node scripts/seedData.js
```

🤝 Development

Key Technologies

- **Frontend:** React 18, TypeScript, Vite, Tailwind CSS, shadcn/ui
- **Backend:** Node.js, Express.js, MongoDB, Mongoose
- **Maps:** MapTiler SDK, Google Maps (fallback)
- **Authentication:** JWT, bcrypt

Development Workflow

1. Start backend: `./start-backend.sh`
2. Start frontend: `./start-frontend.sh`
3. Access frontend at http://localhost:8080
4. API available at http://localhost:3001

🤝 Contributing

1. Fork the repository
2. Create a feature branch (`git checkout -b feature/amazing-feature`)
3. Commit your changes (`git commit -m 'Add 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

Aarjav Jain

- GitHub: @HAWKAARJAV
- Repository: story-swap-locale

🙏 Acknowledgments

- **Shadcn/UI** for beautiful component library
- **Tailwind CSS** for utility-first styling
- **React community** for excellent documentation
- **Vite** for lightning-fast development experience

📞 Support

For support, questions, or feature requests, please open an issue on GitHub or contact the development team.

Happy Storytelling! 🌍 ✨