

final-doc-en-installation-guide

Final Project Documentation: Blog Platform

Introduction

The Blog Platform is a web application designed for publishing and managing articles, announcements, events, and comments. Built with TypeScript, Express, Prisma, PostgreSQL, Next.js, and Mantine, it provides a user-friendly interface and secure backend for user interactions.

Project Goals:

- Create a platform for sharing textual content.
- Support article review workflows for administrators.
- Provide a responsive and intuitive UI.
- Ensure data security and system scalability.

Installation and Setup

Follow these steps to set up and run the project locally:

Prerequisites

- Node.js (version 18 or higher)
- PostgreSQL (version 14 or higher)
- Git

Installation Steps

1. Clone the repository:

```
git clone https://github.com/your-username/blog-platform.git
```

2. Install backend dependencies:

```
cd backend  
npm install
```

3. Set up environment variables (create .env in the backend folder):

```
DATABASE_URL="postgresql://user:password@localhost:5432/blog"  
JWT_SECRET="your-secret-key"
```

4. Run Prisma migrations:

```
npx prisma migrate dev
```

5. Install frontend dependencies:

```
cd ../frontend  
npm install
```

6. Start the backend and frontend servers:

```
cd ../  
./run dev
```

Backend (in backend folder)

```
npm start
```

Frontend (in new terminal, in frontend folder)

```
npm run dev
```

6. Running the Project

- Backend runs at <http://localhost:3000>.
 - Frontend is accessible at <http://localhost:4000>.
 - Visit <http://localhost:4000> to register and test the application.
-

System Architecture

The project consists of two main components:

Backend

- Technologies: TypeScript, Express, Prisma, PostgreSQL
- Router: choose which controller is related for a request
- Controllers: Handle HTTP requests
- Services: project logic
- Models: Defined in schema.prisma

Frontend

- Technologies: TypeScript, Next.js, Mantine
 - Structure:
 - Pages: Dynamic routing (e.g., /articles/:id)
 - Components: Reusable UI with Mantine
-

Usage

- Regular User:
 - Register/login from the homepage.
 - Submit articles or comments from the user dashboard.
 - View public content on the main page.

- Administrator:
 - Access the admin panel at /admin.
 - Review pending articles.
 - Manage announcements, events, and users.

Testing

Unit Tests: Backend tests using Jest (e.g., service layer tests).

cd backend

npm test

- Manual Tests: UI and API testing with Postman.
- Sequence Diagram for article submission:

Deployment

- Server: Deploy backend on Heroku or Vercel.
- Database: Use Neon or Supabase for PostgreSQL.

Frontend: Deploy on Vercel with:

cd frontend

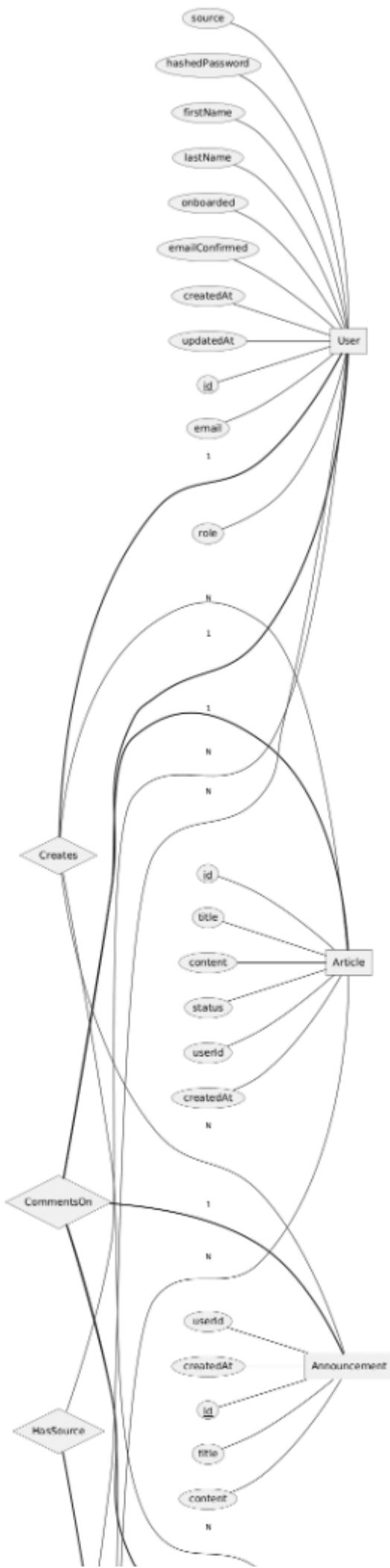
vercel

●

- Data Flow Diagram for deployment:

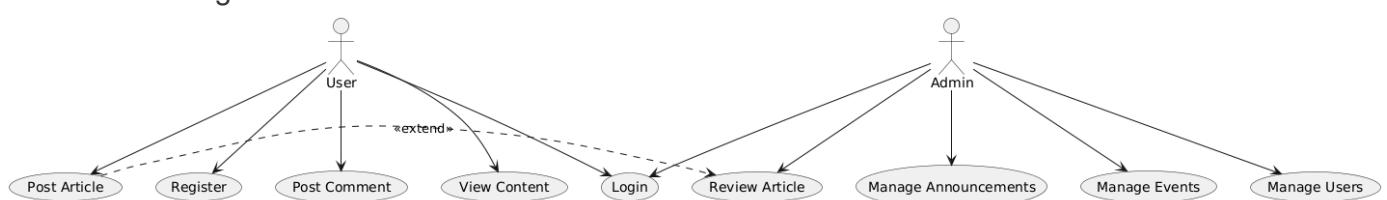
Key Diagrams

- Entity-Relationship Diagram:

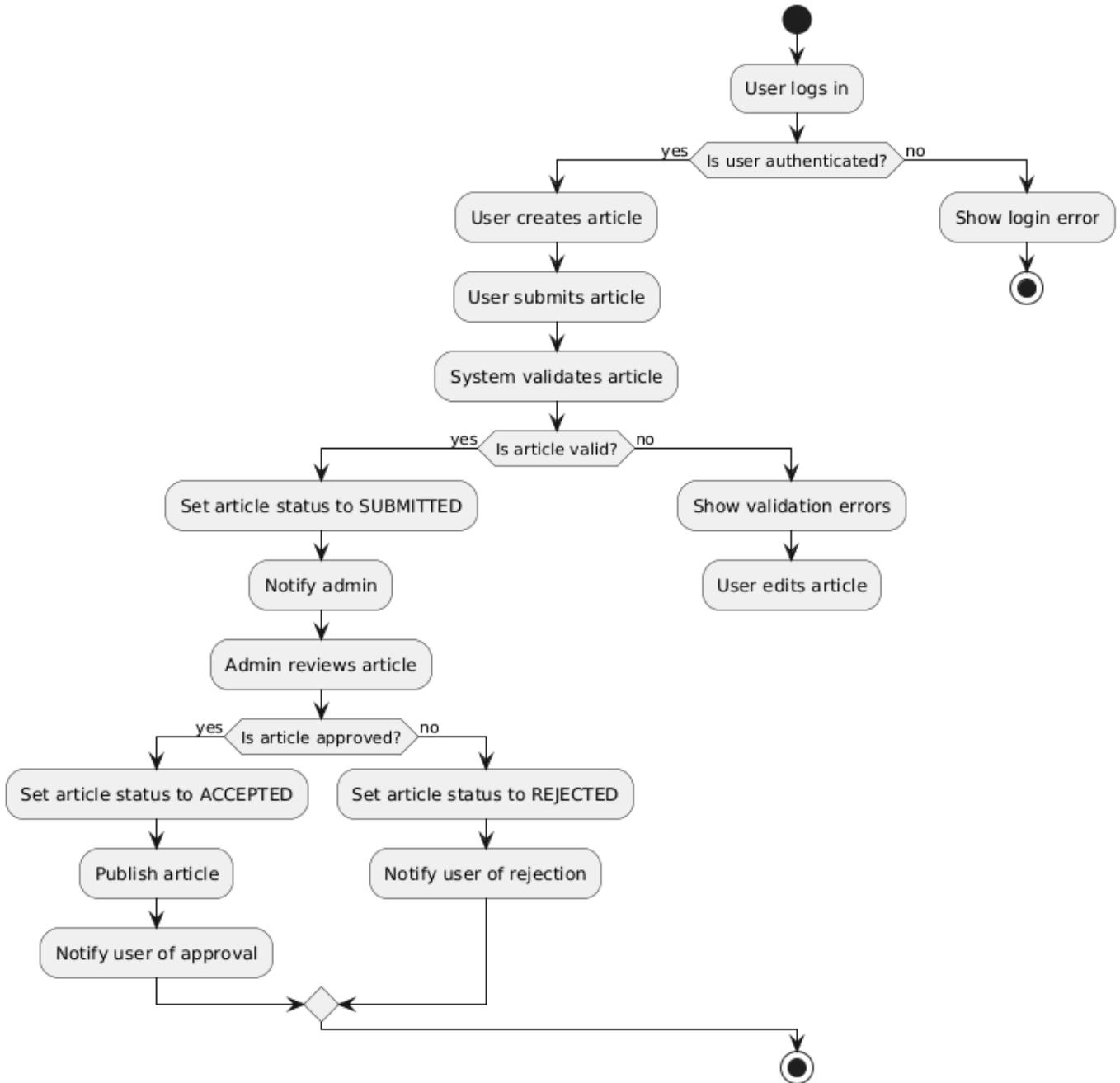




• Use Case Diagram:



- Activity Diagram:



Conclusion

This project delivers a complete blog platform that meets user and admin needs. With a scalable architecture and comprehensive documentation, it's ready for use and further development.

Acknowledgments

Thanks to instructors and teammates for their guidance and support.