

# **Junior Developer Assessment - Blog Dashboard**

## **Overview**

Create a simple blog dashboard that demonstrates understanding of Nuxt 2, Tailwind CSS, and state management fundamentals.

## **1. Project Requirements**

### **Frontend Features**

#### **1. Authentication Pages**

- Login page (`/login`)
- Simple form with email/password
- Redirect to dashboard on success
- Store auth token in Vuex

#### **2. Blog Dashboard**

- Protected route (`/dashboard`)
- Display list of blog posts in a grid
- Each post shows:
  - Title
  - Excerpt (first 100 characters)
  - Created date
  - Status (published/draft)
- Implement pagination (6 posts per page) (*Optional*)

#### **3. Create/Edit Post**

- Form at `/dashboard/post/new` and `/dashboard/post/[id]`
- Fields:
  - Title (required)
  - Content (required)
  - Status (published/draft)
- Save/Update functionality
- Loading states during API calls

## **2. Technical Requirements**

### **Routing & Navigation**

- Implement protected routes using Nuxt middleware
- Create dynamic routes for post editing
- Handle 404 pages gracefully

### **State Management (Vuex)**

- Auth module:
  - Store user token
  - Login/logout actions
  - Persist auth state
- Posts module:
  - CRUD operations
  - Loading states
  - Pagination state

### API Integration

- Use Nuxt Axios module
- Implement interceptors for auth headers (*Optional*)
- Handle API errors gracefully

### UI Requirements

- Responsive design using Tailwind CSS
- Loading states for async operations
- Error messages for form validation
- Toast notifications for success/error messages

### Backend API Specification

API; <https://junior-developer-assessment-blog.onrender.com>

Endpoints	Data Models	Assessment Criteria
<b>POST</b> /api/auth/login <b>GET</b> /api/posts?page=1&limit=6 <b>GET</b> /api/posts/:id <b>POST</b> /api/posts <b>PUT</b> /api/posts/:id <b>DELETE</b> /api/posts/:id	<pre> Post {   id: string   title: string   content: string   status: 'published'   'draft'   createdAt: Date   updatedAt: Date }  User {   email: string   password: string }  // LOG IN {   "email": "test@example.com",   "password": "password" }           </pre>	<ol style="list-style-type: none"> <li><b>Code Quality (40%)</b> <ul style="list-style-type: none"> <li>• Clean, well-organized code</li> <li>• Proper component structure</li> <li>• Effective use of Vuex</li> <li>• Error handling</li> </ul> </li> <li><b>UI Implementation (30%)</b> <ul style="list-style-type: none"> <li>• Proper use of Tailwind utilities</li> <li>• Loading states</li> <li>• Error states</li> <li>• Overall user experience</li> </ul> </li> <li><b>Functionality (30%)</b> <ul style="list-style-type: none"> <li>• All features working as specified</li> <li>• Protected routes functioning</li> <li>• Proper API integration</li> </ul> </li> </ol>

Inspiration: dashboard

Blog Dashboard

+ New Post

[→ Logout

Getting Started with Nuxt

published

Learn the basics of Nuxt.js and how to create your first application...

2024-03-15

Edit

Delete

Mastering Tailwind CSS

draft

Discover the power of utility-first CSS framework and how to...

2024-03-14

Edit

Delete

Previous

1

2

3

Next