
📝 Full Stack Assignment for Internship Hiring

Task Title: Build a Blog Editor Page with Backend & Auto-Save Draft Feature

---

🎯 Objective:

Design and develop a full-stack application that allows users to write, edit, save, and publish blogs with an **auto-save draft feature**.
This task will help us evaluate your skills in **frontend development, backend API design, database integration, and system thinking.**

💻 Requirements:

Frontend:

- Use **React.js, Next.js, Angular, or Vue.js (Your Choice)**.
- Create a **Blog Editor Page**:
  - **Title field** (text input)
  - **Content field** (rich text editor or textarea)
  - **Tags field** (optional, comma-separated)
- Functionality:
  - **Save as Draft button**
  - **Publish button**
  - **Auto-Save Draft (every 30 seconds or when user stops typing for 5 seconds)**
  - Display list of **All Blogs** (published & drafts separately)
  - Edit and update existing drafts/posts

Backend:

- Use **Node.js with Express.js, Django, Flask, or any other framework of your choice.**
- Database: **MongoDB, PostgreSQL, or any SQL/NoSQL database.**
- Define a **Blog schema/model** with fields:
  - `id`
  - `title`
  - `content`
  - `tags`
  - `status` (`draft` or `published`)
  - `created_at`
  - `updated_at`
- API Endpoints (see below)

📡 Sample API Specifications:

Method	Endpoint	Description
POST	/api/blogs/save-draft	Save or update a draft
POST	/api/blogs/publish	Save and publish an article
GET	/api/blogs	Retrieve all blogs
GET	/api/blogs/:id	Retrieve a blog by ID

🎁 Bonus (Optional but preferred):

- Auto-save after **5 seconds of inactivity** using debouncing.
- Notify the user visually when the article is auto-saved (toast, message).
- Use **JWT token or session-based authentication (bonus for protected APIs).**

🗺️ System Architecture Diagram:

```
[ User (Browser) ]
        |
        V
[ Frontend (React/Angular/Vue) ]
        |
  REST API Calls (HTTP)
        |
        V
[ Backend (Express/Django/Flask API Server) ]
        |
        V
[ Database (MongoDB/PostgreSQL) ]
```

✅ Evaluation Criteria:

Skillset	Evaluation Area
Frontend	UI, Form validation, State management, Auto-save UX
Backend	Clean API design, Data validation, Error handling
Database	Schema design, Efficiency
System thinking	Modular code, Clean architecture, Clear separation of concerns
Bonus points (optional)	Auto-save optimizations, Notifications, Authentication

🔗 Deliverables:

- **GitHub Repo with clear README** (Setup instructions + tech stack)
- Working **Demo (optional) or video walk-through**
- Clean code and comments.