
Frontend Developer Intern — Shortlisting Assignment (Auth + Dashboard) : Estimated Time : 2 hours

About the assignment

We're shortlisting **Frontend Developer Interns** (final-year students / recent grads) who can build modern UI and integrate it with a basic backend.

Deadline: 3 days from receiving this assignment

Expected effort: ~2 hours (time-boxed)

Goal

Build a small web app with:

- ✓ **Frontend (primary):** Auth + dashboard + CRUD UI
 - ✓ **Backend (supportive):** minimal APIs + DB to support frontend integration
-

✓ Must-have requirements

A) Frontend (Primary Focus)

1. Build using **React.js** or **Next.js**
 2. Responsive UI using **TailwindCSS** / **Material UI** / **Bootstrap**
 3. Forms with validation:
 - Client-side validation (required fields, email format, password rules)
 - Display server-side error messages clearly
 4. Protected routes:
 - Dashboard accessible only after login
 5. Good UX basics:
 - Loading states, error states, success messages
-

B) Backend (Supportive / Minimal)

Use **Node.js + Express** OR **Python (FastAPI/Django)**

Implement APIs for:

1. **Auth**
 - Signup + Login
 - Password hashing (bcrypt/argon2)
 - JWT auth middleware
 2. **Profile**
 - Fetch profile (GET /api/v1/me)
 - Update profile (PUT /api/v1/me)
 3. **CRUD entity** (choose ONE): **tasks** OR **notes** OR **posts**
 - Create, Read (list + single), Update, Delete
 4. **Database**: MongoDB / Postgres / MySQL
 5. API versioning: /api/v1/... + consistent error responses
-

C) Dashboard Features (Frontend)

1. Show user profile (from backend)
 2. CRUD UI for chosen entity
 3. Search + filter UI for list page
 4. Logout flow
-

Security & code quality (Must-have)

- Password hashing (no plain text passwords)
- JWT validation on protected routes
- Input validation (backend) + clear error messages
- Project structured for easy scaling (separate modules/components)
- Basic logging + error handling (backend)

Bonus (optional): Docker, unit tests, refresh tokens, pagination, rate limiting.

Deliverables (What to submit)

1. **Single GitHub repo** containing frontend + backend
2. **README.md** must include:

- Tech stack
 - Setup steps (env vars + DB steps)
 - How to run frontend + backend
 - 2–3 demo credentials or seed steps (optional)
3. **Postman collection** OR Swagger/OpenAPI
 4. Short note (5–10 lines): “How would you scale this for production?” (deployment, CORS, env management, DB indexing, caching, etc.)
-

Evaluation criteria (How we score)

- UI/UX quality + responsiveness
 - Frontend–backend integration quality (auth + CRUD)
 - Security practices (hashing, JWT protection, validation)
 - Code structure + cleanliness (modularity, naming, reuse)
 - Documentation + reproducibility (README quality)
 - Scalability potential (folder structure, patterns)
-

HOW TO APPLY

Email your **resume + GitHub link** to:

- joydip@primetrade.ai
 - hello@primetrade.ai
 - chetan@primetrade.ai
- CC: sonika@primetrade.ai
-