🚀 Full Stack Developer Intern – Assignment

Dear Candidate,

As part of the hiring process for the **Full Stack Developer Intern** position, we want to evaluate your problem-solving, coding, and full-stack development skills.

You are required to complete **at least ONE** of the following tasks (you may attempt both if you wish).

> 🔑 **Bonus Points** if you:
>
> - Deploy/host the app and share a working link.
> - Add extra features or polish.
> - Submit both tasks instead of one.

If hosting is not possible, please share a **screen-recorded video demo** of your project along with the code.

📌 Task 1: Real-Time Voting App

Objective

Build a simple voting app where multiple users (from different browsers) can log in and cast votes.

Requirements (Core)

1. **Login Page** – User enters a name (no password needed).
2. **Voting Page** – Show 2–3 options (e.g., Option A, Option B, Option C).
3. Each user can cast only **one vote** per session.
4. **Results Page** – Show live tally of votes (even if updated only on manual refresh).

Stretch Goals (Optional)

- Add **real-time updates** using WebSockets / Server-Sent Events.

- Display results as a **bar chart or pie chart**.
- Prevent duplicate votes from the same session.

🔥 Challenge

- Add **auto-refresh** for results every 5 seconds (if real-time not done).
- Or, implement **auto reload** in the browser on any vote update.

📌 Task 2: Mini Helpdesk App

Objective

Build a small ticketing app where users can raise tickets and admins can manage them.

Requirements (Core)

1. **Ticket Form** – User submits (Name, Issue, Priority).
2. **Dashboard** – Display all open tickets in a list/table.
3. **Status Update** – Allow admin to change ticket status (Open / In Progress / Closed).
4. Store tickets in a database (SQL/NoSQL).

Stretch Goals (Optional)

- Auto-refresh dashboard every 5 seconds (or use WebSockets for real-time updates).
- Add filters (e.g., by priority or status).
- Show ticket createdAt time and sort by latest.

🔥 Challenge

- Add **auto-refresh** for dashboard tickets.
- Or, implement **auto reload** in browser when a ticket is updated.

✅ Submission Guidelines

- You must complete **at least 1 task**. Completing both will fetch bonus points.
- Share your code via:
  - GitHub / GitLab repo link, or
  - Zipped project folder.
- Along with code, submit:
  - A **screen-recorded demo video** showing the working app, OR
  - A **live hosted link** (preferred).
- Ensure your app runs with a simple `npm install && npm start` (or equivalent).
- Mention any assumptions or limitations in a README file.

🎯 Evaluation Criteria

- **Core functionality** completion.
- **Code quality** (clean, structured, modular).
- **Frontend + Backend + Database** integration.
- **UI/UX** (simplicity and usability).
- Bonus for **hosting / extra features / stretch challenges**.

👉 At least **one task is mandatory**. Feel free to be creative and go beyond requirements if time permits.

Good luck & happy coding 🚀