

## Full Stack Developer Evaluation Task

# WhatsApp Web Clone

Your mission If you choose to accept, is to build a **WhatsApp Web – like chat interface** that displays real-time WhatsApp conversations using webhook data. The interface should mimic WhatsApp Web, show chats neatly, and support sending new messages (for storage only – no external sending).

---

### You Will Need to:

- Set up a **MongoDB Cluster** (preferably MongoDB Atlas)
  - Create **server-side APIs** in **Node.js** or **Python** to process payloads and serve data
  - Build a **responsive, mobile-friendly frontend** that looks and behaves like WhatsApp Web
  - Host your application on a **public URL** (e.g., using Vercel, Render, Heroku, or your own server)
- 

### Task 1: Webhook Payload Processor

- Here is a zip file with sample payloads (JSON files) that simulate WhatsApp Business API webhooks:  
Download It here: [https://drive.google.com/file/d/1pWZ9HaHLza8k080pP\\_GhvKl8j2voy-U/view?usp=sharing](https://drive.google.com/file/d/1pWZ9HaHLza8k080pP_GhvKl8j2voy-U/view?usp=sharing)
- Write a script to:
  - Read these payloads
  - Insert new messages into the MongoDB collection:  
**Database:** whatsapp  
**Collection:** processed\_messages
  - Use the **status payloads** (sent, delivered, read) to **update the status** of each corresponding message using the id or meta\_msg\_id field

### Task 2: WhatsApp Web-Like Interface

- Create a **frontend UI similar to WhatsApp Web**
- Show all **conversations** grouped by user (wa\_id)
- Clicking a chat should show:
  - All message bubbles with date time
  - Status indicators (sent, delivered, read)
  - Basic user info (name and number)

- Design should be:
    - **Clean**
    - **Responsive**
    - **Easy to use on mobile and desktop**
- 

### Task 3: Send Message (Demo)

- Add a “**Send Message**” input box like WhatsApp Web
  - On submit, the message should:
    - Show up in the conversation UI
    - Be saved to the database (processed\_messages)
    - No message will actually be sent outside the platform
- 

### Task 4: Deployment

**Host the complete app on a public URL** You may use: Vercel, Render, Heroku, Or any other hosting provider

- **Do not submit code-only repositories or localhost URLs**
- Demo must be accessible publicly without setup

### Bonus Task: Real-Time Interface Using WebSocket (Not Mandatory)

Simulate real-time message updates using **WebSocket** (e.g., Socket.IO). Your frontend should automatically reflect new messages and status updates as they are inserted or updated in the database, without requiring a manual refresh.

### Evaluation Criteria

- Closeness of your app UI to WhatsApp web
- Responsiveness on mobile devices
- Thoughtful assumptions and attention to detail
- A well-structured backend
- Responsiveness and mobile friendliness

### Submission Instructions:

Submit ONLY using the [MakeList Form here](#). Submissions via email or internshala chat won't be evaluated.

### Requirements

- Submit only if you're:
  - Based in **India**

- **Not** a student, intern, or part-time applicant
- Final Submission Must Include:
  - **Public URL to the running app**
  - (Optional) GitHub repo or documentation

---

## NOTES

- No real WhatsApp messages will be sent. This is a frontend + backend simulation for webhook data processing and display.
- Use of AI based code generation is acceptable as long as you are able to explain it