



Manvik Talwar &lt;manvik.talwar@gmail.com&gt;

---

## Subject: Backend Round 1 Task – QOTD API | TechLearn Solutions

---

**TechLearn Solutions** <techlearnsolutions@gmail.com>

Sat, Jan 31, 2026 at 8:00 AM

To: Shrashiti Dwivedi <shrashtidwivedi91@gmail.com>, Venkata Pavan Kumar Utpala <uvpavankumar2006@gmail.com>, Hanish Gori <gorihanish@gmail.com>, Ashmit Aggarwal <ashmit12515@gmail.com>, Adith Nambiar <adithnambiar27@gmail.com>, Manvik Talwar <manvik.talwar@gmail.com>, Priyanshu Saini <priyanshusaini124@gmail.com>, Yashovardhan Gupta <gyashovardhan@gmail.com>, aayushi sabharwal <aayushi.sabharwal@gmail.com>, Jeevan Kumar <jeevank6699@gmail.com>, archit.kushwaha777@gmail.com, Nikunj Gupta <nikunj.gupta17072005@gmail.com>, project.samarops@gmail.com, nishant.yadav.3457@gmail.com, Ashmit Agrawal <ashagrawal001@gmail.com>, onlyvihan22@gmail.com, Aditya Bansal <adityabansal6404@gmail.com>, Chaitanya Vishwakarma <chaitanyavish2411@gmail.com>, Tejas Kharche <tejaskharche03@gmail.com>, shivangigct@gmail.com, Akshit Saini <sainiakshit1502@gmail.com>, yandurisaitanush@gmail.com, Anushka Srivastava <anushkasri0904@gmail.com>, amarnanimansi04@gmail.com

Hey there Candidate,

As part of **Round 1 of the interview process**, we'd like you to work on a **Backend-focused technical task**. This task is designed to evaluate your backend fundamentals, API design, data modelling, problem-solving ability, and communication.

---

## Task: Build Backend for QOTD (Question of the Day)

### Task Overview

You are required to **design and implement the backend logic and APIs** for a **QOTD (Question of the Day)** feature for an edtech platform such as **TechLearn Solutions**.

The QOTD represents a **daily DSA coding challenge** attempted by students globally. Each day, a single question is shown, students submit solutions, and their performance is evaluated.

---

### Functional Requirements

Your backend should support the following core features:

#### 1 QOTD Management

- API to fetch today's question

Each question should include:

- Title
- Difficulty (Easy / Medium / Hard)
- Problem statement
- Sample input & output
- Expected output / solution (can be hidden)

#### 2 Submission & Evaluation

- API to submit a user's answer
- Mock or basic evaluation logic is acceptable
- Example: match expected output, predefined test cases, or dummy validation
- Return a clear response (correct / incorrect / partially correct)

### 3 Supporting Features (Design-Level)

Design APIs or data structures for:

- Hints
  - Statistics (attempt count, success rate)
  - Leaderboard (top scores / fastest submissions – mock data allowed)
- 

### Data & Tech Stack

- Backend stack of your choice (Node.js preferred but not mandatory)
  - Database can be: In-memory
  - JSON
  - MongoDB / PostgreSQL (optional)
  - No authentication required
  - Focus on **clean structure and scalability**, not completeness
- 

### API Expectations

- RESTful API design
  - Proper status codes
  - Basic error handling
  - Clear and readable code structure
- 

### Deployment (Mandatory)

Please deploy your backend using any platform such as:

- Render
  - Railway
  - Fly.io (or similar)
- 

### Submission Requirements

Please submit:

- GitHub repository link
  - Live deployed API base URL
  - README.md including: Tech stack used
  - API endpoints with sample requests/responses
  - Data model explanation
  - How to run the project locally
  - What you would improve with more time
- 

### Deadline

**24 hours** from the time you receive this email.

You may submit the task **anytime before the deadline** once completed. There is no need to rush—take your time to design and implement it thoughtfully.

 Please note: Candidates who do not submit the task on time or submit an incomplete task

