Full Stack Developer Test Kit

Duration: Weekend (approximately 12-15 hours total effort)

---

1. Guidelines for the Candidate

- You are required to complete the test within the given timeframe (the weekend).
- Feel free to use any full-stack tech stack of your choice (e.g., React, Angular, Bootstrap, Node.js, Django, etc.).
- Ensure that your code is clean, modular, and well-documented.
- Focus on functionality, design, and responsiveness.
- Submit the project code, a deployment link (if possible), and a short README file with instructions to set up the project.

---

2. Problem Statement

Build a Mini Full-Stack Web Application

Scenario:
Imagine you're working for a startup that wants to launch a **Task Manager Application**. Your job is to develop a **CRUD-based web application** where users can:

1. **Sign up and Log in** (Authentication).
2. Create, read, update, and delete tasks (CRUD operations).
3. Assign categories or tags to tasks.
4. Mark tasks as completed or pending.
5. View tasks by category or completion status.
6. Search tasks by title.

---

3. Technical Requirements

Frontend Requirements:

- Create a user-friendly interface for task management.
- Make the application responsive (mobile and desktop).
- Include form validation (e.g., for sign-up and task creation).

Backend Requirements:

- Build an API to handle all backend operations.
- Implement secure user authentication (e.g., JWT or session-based).
- Use a database of your choice (e.g., PostgreSQL, MongoDB, SQLite).

Optional Enhancements (Bonus):

- Add a priority level (High, Medium, Low) for tasks.
- Implement a due date reminder using notifications.
- Include a dashboard that shows a summary (e.g., total tasks, pending tasks, completed tasks).
- Deploy the app on any platform (e.g., Vercel, Netlify, Heroku).

---

4. Key Evaluation Criteria

Candidates will be evaluated on:

1. **Functionality**: Does the app meet the basic requirements (CRUD operations, authentication)?
2. **Code Quality**: Is the code modular, clean, and follows best practices?
3. **User Experience**: Is the app visually appealing and easy to use?
4. **Problem-Solving Skills**: How well does the candidate implement features and handle edge cases?
5. **Documentation**: Does the submission include clear instructions to run the project?
6. **Creativity**: Bonus points for adding optional enhancements or unique features.

---

5. Submission Details

Candidates must submit the following:

1. A GitHub repository link containing the code.
2. A deployment link to test the app (optional but recommended).
3. A README file with the following details:
    - How to set up and run the project.
    - Any assumptions made during development.
    - List of technologies/libraries used.
    - Challenges faced and how they were addressed.

---

6. Sample Evaluation Template (For Recruiters)

Criteria	Score (0-10)	Comments
Functionality		
Code Quality		
User Experience (UI/UX)		

Problem-Solving Skills

Documentation

Creativity/Enhancements