## 📄 Project Brief

### Project Name: **Testify** – Online Testing Platform

### Description:

Testify is an online testing and assessment system that allows users to take multiple-choice or essay-based tests. The platform enables test creators to build customizable tests, analyze performance, and store user answers and test results.

### Target Users:

* Test takers (students, learners)
* Test creators (teachers, HR staff)
* Admins

### Goals:

* Allow users to take tests and view detailed results
* Allow test creators to manage questions and create test sets
* Secure and scalable architecture

### Scope:

* Web-based platform
* Multi-user roles: user, admin, test creator
* Test creation, submission, and analytics

## 📝 Functional Requirements

1. **User Management**
   * Sign up, login, logout
   * Role-based access: user, admin, creator
2. **Test Management**
   * Create/update/delete tests
   * Add questions and sections
   * Assign difficulty, time limit, etc.
3. **Question Management**
   * Create/update/delete questions
   * Support multiple correct answers
   * Attach images to questions and answers
4. **Test Participation**
   * Take tests (timer included)
   * View questions with images and answers
   * Submit and receive immediate results
5. **Submission & Results**
   * Store submission history per user
   * Show detailed feedback (correct/wrong, score, explanation)
   * View past submissions
6. **Admin Panel**
   * Manage users
   * Manage roles
   * View system stats

## 🛠 Non-functional Requirements

* **Performance**: Load test in < 1s for <100 concurrent users
* **Security**: JWT authentication, hashed passwords
* **Scalability**: Modular service structure
* **Availability**: 99.9% uptime
* **Responsiveness**: Optimized for mobile & desktop

## 📐 System Architecture Overview

* **Frontend**: ReactJS or Next.js
* **Backend**: Node.js (Express) or Spring Boot (Java)
* **Database**: PostgreSQL
* **Auth**: JWT, Role-based access
* **Deployment**: Docker + Railway/Heroku (Dev) | VPS (Prod)

## 🎨 UI/UX Mockup Plan (Textual Description)

### Pages:

1. **Home Page** – Welcome message, list of available tests
2. **Login / Register** – Auth forms
3. **Dashboard** – My tests, past results
4. **Test Viewer** – Question list, timer, submit button
5. **Results Page** – Score summary + detailed answers
6. **Admin Panel** – User and test management interface

Design with Figma or similar tools. Prioritize clean, minimal layout.

## 📌 API Spec (Outline)

### Auth

* POST /auth/register
* POST /auth/login

### Users

* GET /users/:id
* PUT /users/:id

### Tests

* GET /tests
* POST /tests
* GET /tests/:id

### Questions

* POST /questions
* GET /questions/:id

### Submissions

* POST /submissions
* GET /submissions/:id
* GET /users/:id/submissions

Use Swagger or Postman for actual implementation.

## ✅ Notes

* Extend features to include PDF export of test results
* Add email notifications in future phase
* Consider real-time collaboration/test-taking with sockets