

Online Aptitude Test Platform with Timed Mocks and Analytics

Project Abstract · Web Development · Built-to-order final-year project · projectech.in

1. Title

Online Aptitude Test Platform with Timed Mocks and Analytics — a web-based platform that helps engineering students prepare for placement aptitude rounds through timed mock tests, an untimed practice mode, a searchable question bank, and a results dashboard with section-wise accuracy, time analysis, percentile ranking and full worked-solution review.

2. Introduction / Background

Aptitude tests are the first elimination round in most campus placement drives, yet students typically prepare from static PDFs and question lists with no sense of time pressure or performance tracking. Existing free resources rarely combine a realistic timed environment with analytics: students cannot tell whether they are losing marks to weak concepts, slow solving, or poor time allocation across sections. This project builds a complete web platform that reproduces the pressure of a real test — a countdown timer with automatic submission — and then converts each attempt into actionable analytics: score, accuracy per section (quantitative, logical reasoning, verbal), average time per question, a percentile against a sample cohort, and a question-by-question review where every answer is paired with its worked solution.

3. Problem Statement

Placement-aspiring students lack an affordable, self-hosted way to (a) practise aptitude questions under genuine time constraints, (b) measure section-wise strengths and weaknesses quantitatively, and (c) review mistakes with step-by-step solutions instead of bare answer keys. Generic quiz plugins provide none of the placement-specific workflow: timed auto-submit, per-question timing, flag-for-review, or cohort percentile. The project addresses this gap with a purpose-built aptitude testing platform whose question bank, timer engine and analytics are designed around the placement-preparation use case.

4. Objectives

- Provide a bank of 36 original aptitude questions (12 quantitative, 12 logical reasoning, 12 verbal), each with four options, one correct answer, a difficulty tag and a worked solution.
- Implement a timed mock mode: 20 questions in 20 minutes, live countdown, question palette with answered/flagged states, and automatic submission at zero.
- Implement an untimed practice mode with instant feedback and immediate worked-solution display.
- Generate a results dashboard: overall score, section-wise accuracy bars, average and total time, skipped count, percentile vs a built-in sample cohort, and full question review.
- Provide a filterable question-bank browser (by section, difficulty and keyword) with expandable solutions.
- Keep the entire front end in a single deployable file with no build step, so the ordered full-stack build can adopt it directly.

5. Proposed Methodology

The question bank is authored as a structured data array where each entry carries the question text, four options, the correct-option index, section, difficulty and a worked solution string — keeping content fully separated from presentation so the bank can grow without code changes. The mock engine is a finite-state flow (idle → running → submitted): on start it loads a fixed 20-question paper (7 quantitative, 7 logical, 6 verbal), starts a one-second interval timer, and accumulates per-question dwell time on every navigation event. The palette is re-rendered from a single state object, guaranteeing the answered/flagged/current indicators can never disagree with the underlying data. On submit (manual or timer-expiry), a grading function compares selections against the bank's answer keys, aggregates per-section tallies, and computes the percentile by ranking the score against a deterministic 240-entry sample cohort generated with a seeded pseudo-random distribution. Practice mode reuses the same bank and option-rendering components with the timer disabled and feedback shown inline. The bank browser is a pure filter over section, difficulty and keyword with lazy solution reveal.

6. System Architecture / Working

The prototype is a single-page application with hash routing across five views: Home, Timed Mock, Practice, Question Bank and Results. A top navigation bar persists across views. The Timed Mock view splits into the question card (question text, four options, previous/next, flag-

for-review) and a sidebar (countdown timer, 20-cell question palette, submit button). State lives in one in-memory object per attempt; the timer interval updates both the display and the accumulated per-question times. The Results view renders a score hero, three section-accuracy bars, a percentile explainer panel, and a review list where each entry shows the student's answer, the correct answer, time spent and the worked solution. The built-to-order full-stack version extends this architecture with user accounts and authentication, a database-backed question bank with an admin panel for adding/editing questions, persistent attempt history, and a real-time backend — the front-end prototype's components map one-to-one onto these services.

7. Hardware / Software Requirements

Item	Requirement
Client device	Any laptop/desktop with a modern web browser (Chrome, Edge, Firefox, Safari); approximately 2 GB free RAM (typical)
Front-end (prototype)	HTML5, CSS3, vanilla JavaScript (ES6) — no frameworks, no build step
Full-stack build (ordered version)	Node.js + Express (or equivalent) backend, PostgreSQL/MySQL database, JWT-based authentication, REST API between the front end and the question/attempt services
Question bank	36 original questions shipped in the prototype; admin panel in the ordered version for unlimited additions
Hosting (ordered version)	Any VPS/shared host with Node.js support; static hosting suffices for the prototype

8. Expected Outcomes

- A working timed mock: 20 questions, a 20-minute countdown that auto-submits at zero, per-question timing, and a flag-for-review palette.
- A results dashboard showing score, section-wise accuracy, average time per question, skipped count, percentile against the built-in sample cohort, and a complete solution review.
- A practice mode with instant correct/incorrect feedback and worked solutions, plus a question-bank browser filterable by section, difficulty and keyword.

- Clean, original question content: every one of the 36 questions carries a verified worked solution.
- Complete documentation: this abstract, a setup guide, and viva Q&A covering the timer engine, grading logic, percentile computation and full-stack extension plan.

All performance figures are design targets for the ordered build; no measured claims are made. The cohort percentile in the prototype compares against illustrative sample data, stated as such in the UI.

9. Applications

- Campus placement cells can host it as their internal aptitude-practice portal.
- Coaching institutes can use the timed-mock + analytics flow for batch assessments.
- Students get a realistic, self-paced placement preparation tool with genuine feedback on weak areas.
- Colleges can extend it into a proctored online-exam module (listed under Future Scope).

10. Future Scope

- Adaptive testing: difficulty adjusted per student based on running accuracy.
- Proctoring aids: tab-switch detection, webcam snapshot checks, full-screen enforcement.
- Larger, community-curated question bank with admin moderation and bulk import.
- Leaderboards, streaks and study plans driven by historical attempt data.
- Mobile application reusing the same question-bank API.
- Multi-language question content and accessibility improvements (screen-reader labels, keyboard navigation).