Product/Project Outline Document (POD)

Mock Test & Grand All- India Test Platform (LMS- Style)

Tech Stack: Php and laravel

---

1. Project Overview

A full-featured online examination system for:

- **Mock Tests** (monthly, category-wise)
- **Grand All- India Tests** (fixed date, same time for all students)

Students subscribe or purchase access, take tests within scheduled windows, get results instantly, and view national ranklists for grand tests. Admins can upload questions (individually or bulk), create tests, schedule them, randomize questions per student, and manage analytics.

---

2. User Roles & Permissions

Instructor (New Role)

Instructors are assigned by Admin to specific students or categories. They can:

- View assigned students
- See student queries/doubts
- Respond via an internal **chat system**
- Track categories they manage (Mains, Groups, etc.)
- Guide students on exam preparation, concepts, mistakes

New Feature: Student–Instructor Chatbot System

- Students can raise doubts from their dashboard
- Doubts can be:
- Text messages
- Screenshot/images
- System assigns the doubt automatically based on **category** or manually by Admin
- Instructor receives notification
- Instructor replies through a real-time chat interface
- Students see all replies in their chat module
- Chat history stored per student

- Admin can monitor all chats
- Optional AI-chatbot to provide instant automated replies to common doubts

2. User Roles & Permissions (Existing)

Admin

- Create, edit, delete **questions**
- Bulk upload questions via **CSV/Excel/JSON**
- Create and configure **Mock Tests** and **Grand Tests**
- Define **randomization rules**
- Set **availability windows (start/end)**
- Publish/Unpublish tests
- View results, ranks, and analytics

Student

- Register / Login / Profile
- Subscribe or purchase tests
- Access tests within availability windows
- Take tests with individual timer
- View instant results
- View Grand All- India top ranks

3. Core Concepts

3.1 Availability Window (Test-Level Timing)

Each test contains a **global open and close time**:

```
available_from

available_to
```

This defines **when** the test can be started. Outside the window → student cannot start the test.

3.2 Duration (Per- User Timer)

Once a student clicks **Start Test**, their **personal timer** begins. Example:

- Test window: 10 AM → 1 PM
- Duration: 90 minutes

- Student starts at 11:00 → ends at 12:30
- If student starts at 12:30 → ends at 1:00 (forced auto-submit)

3.3 Mock Test vs Grand Test

Mock Tests

- Rolling window (long period like 1 day or 30 days)
- Students can start anytime in the window
- Duration applies individually

Grand All- India Tests

- Fixed window (same date/time for all students)
- Competitive environment
- National ranking generated after submissions

---

4. Question Management

4.1 Creating Questions Individually

Admin fills a form:

- Title
- Options
- Correct answers
- Marks & negative marks
- Difficulty (easy/medium/hard)
- Category (Mains, Groups, etc.)
- Tags

Stored in MongoDB as:

```
{
  _id: "q123",
  title: "Which is a prime
  number?", type: "mcq-single",
  options: ["2","3","4","6"],
  correct: [0],
  marks: 2,
  negative: 0.5,
  tags: ["number-
}
```

4.2 Bulk Upload Questions (CSV / Excel / JSON)

Workflow

1. Admin uploads file
2. Backend parses using libraries like **xlsx** or csv-parser
3. Questions mapped to schema format
4. Validation applied (correct index, fields present)
5. Bulk inserted using `insertMany(`
6. Error report generated for invalid rows

Example CSV

```
title,options,correct,marks,negative,difficulty,categor
y   What is 2+2?,3;4;5;6,1,1,0,easy,Maths
Capital of India,Delhi;Mumbai;Kolkata;Chennai,0,1,0,easy,GK
```

---

5. Test Creation Flow (Admin)

Step-by-Step:

Step 1: Basic Test Info

- Title
- Description
- Category (Mains/Groups/etc.)
- Type (Mock / Grand)
- Duration (in minutes)
- Negative marking (yes/no)

Step 2: Add Questions

Admin selects:

1. **Auto Selection (Random)** – define rules:
2. 10 Easy
3. 10 Medium
4. 5 Hard

Backend uses Mongo`$sample` to pick random questions.

Step 3: Randomization Settings

- Shuffle question order
- Shuffle options per question
- Random question selection (unique set per student)

Step 4: Availability & Rules

- `available_from` datetime
- `available_to` datetime
- For Grand Tests → fixed same window for all
- For Mock Tests → flexible window
- `perUserStart:`

Step 5: Publish Test

Once published:

- Students see it in their dashboard
- But can start **only within availability window**

6. Question Selection & Randomization Logic

This section explains how questions are selected from the Question Bank.

Admin no longer selects individual questions. Reason:

- Ensures fairness
- Prevents bias
- Automates large test creation
- Maintains consistent difficulty

6.1 Auto Selection (Random Based on Rules)

Admin defines:

- Number of easy / medium / hard questions
- Subject
- Category

Backend uses SQL `$sample`:

```
await Question.aggregate([
  { $match: { category: "Maths", difficulty: "easy" } },
  { $sample: { size: 10 } }
])
```

Each test is built automatically.

6.2 Random Per Student

Each student receives:

- Unique set of questions
- Random question order
- Random option order

This prevents cheating and repetition.

6.3 Method 3 — Random Per Student (Unique Paper for Each Student)

When admin enables:

```
randomPerStudent: true
```

Then for **each student**, when they click *Start Test*:

- System fetches a **fresh random set** from the Question Bank
- Applies rules defined by admin
- Creates a personal question list for that

student This means:

- Student A and Student B get **different questions**
- Even if they attempt the same

test Useful for:

- Preventing cheating
- High-volume mock exams
- Continuous learning practice tests

6.4 Random Option Shuffle

To add more security:

- Each question's options are shuffled per student
- Correct answer index is stored in the Attempt

record This ensures no two students get the same

order.

---

*(*7. Student Test Flow

1. Student logs in
2. Goes to Dashboard
3. Sees upcoming/active tests
4. Clicks **Start Test** (only allowed inside availability window)
5. Backend:
6. Creates Attempt document
7. Applies randomization
8. Starts timer
9. Student answers questions
10. Auto-save every 10–30 seconds
11. On submit:
12. Backend grades instantly (for objective questions)
13. Saves results
14. Student sees result page with:
15. Score
16. Correct/Incorrect
17. Section analysis
18. Time spent

---

8. Ranking (Grand All- India Tests)

After all submissions:

- Scores sorted
- Rank assigned
- Top 3 specially highlighted
- Ranklist available to all students

9. Summary

This system allows:

- Flexible and fixed exam scheduling
- Individual or bulk question creation
- Random tests for every student
- Instant grading and national ranking
- Full control for admin and seamless experience for students