

SMART TEST CENTER — Architectural Diagram

High-level Component Diagram

```
Client (React Frontend — Students / Mentors / Admin)
|
Server (Node.js / Express API)
- Auth Controller (JWT + OAuth)
- Test Controller
- AI Question Generator Service
- Proctoring Service (Webcam + Anti-cheat)
- Assignment Controller
- Notification Controller (Socket.io / REST)
- Result & Grading Service
- File Upload Handler (GridFS/local FS)
|
Data (MongoDB Compass + File Storage)
|
Realtime (Socket.io Server)
```

Collections / Data model

```
USERS: name, email, phone, role, passwordHash, photoUrl, organization, experience, yearOfStudy
TESTS: title, type, duration, totalQuestions, mentorId, createdAt, randomized
QUESTIONS: testId, type, text, options, answer, codingSpec
ASSIGNMENTS: title, fileUrl, mentorId, dueDate
RESULTS: testId, studentId, score, answers, submittedAt, autoSubmitted, warnings
NOTIFICATIONS: message, senderId, recipientIds, createdAt
```

Feature-specific component details

- Authentication: JWT + OAuth (Google/Facebook)
- Test creation: MCQ / Long / Coding + AI generated questions
- Proctoring: Anti-cheat, webcam, tab-switch detection, auto-submit after warnings
- Test-taking: Randomized questions, timer, coding editor (Monaco), run/submit
- Notifications: Real-time with Socket.io, fallback to REST
- Assignments: Mentor uploads/downloads with GridFS/local FS
- Results: Auto-graded MCQ/coding, manual/AI-assisted long answers

Sequence Flow

```
Mentor -> Create Test (optionally AI generate)
Backend -> Saves test & questions in MongoDB
Student -> Takes Test (randomized questions)
Backend -> Grades & saves results
Student -> Views Results & Detailed Report
```

Local Development Setup

```
Frontend: http://localhost:3000
Backend: http://localhost:5000
MongoDB: mongodb://localhost:27017/smart-test-center
Socket.io: same as backend
```

Security & Privacy Notes

- Passwords hashed with bcrypt
- HTTPS in production
- Webcam data handled with consent and retention limits
- Rate limiting for login and AI generation endpoints

