Great! Let’s now go through the **Test Plan and Test Cases** to ensure your application is reliable, bug-free, and production-ready.

**📄 7. Test Plan & Test Cases**

**Project:** *FreelancerConnect*  
**Prepared By:** [Your Name]  
**Testing Types:** Unit, Integration, UI/UX, Security, UAT (User Acceptance Testing)

**📘 1. Test Plan Overview**

**🧪 Objectives:**

* Verify that all features work as expected.
* Identify and fix bugs before deployment.
* Validate the UI/UX and responsiveness across devices.
* Ensure secure handling of authentication and payment.

**🧷 2. Scope of Testing**

| **Module** | **Test Type** |
| --- | --- |
| User Registration/Login | Functional, Security |
| Project Posting | Functional |
| Proposal Submission | Functional, Validation |
| Messaging | Integration, UX |
| Payment Gateway | Integration, Security |
| Admin Controls | Role-based, Functional |
| API Performance | Load, Stress Testing |

**🧪 3. Testing Types**

* **Unit Testing** – Using JUnit/TestNG for backend methods.
* **Integration Testing** – Testing API + database + external service interaction.
* **UI/UX Testing** – Testing navigation, responsiveness, accessibility.
* **Security Testing** – JWT handling, role protection, SQL injection, etc.
* **Regression Testing** – After bug fixes.
* **User Acceptance Testing (UAT)** – Validating flow with mock users.

**🧾 4. Sample Test Cases**

**✅ TC-001: User Registration (Freelancer)**

| **Test Case ID** | **TC-001** |
| --- | --- |
| Description | Register new freelancer with valid data |
| Input | Name, Email, Password, Role=Freelancer |
| Expected Output | Success message + JWT |
| Status | Pass/Fail |

**✅ TC-002: User Login (Invalid Credentials)**

| **Test Case ID** | **TC-002** |
| --- | --- |
| Description | Try logging in with wrong email/password |
| Input | Email: [wrong@example.com](mailto:wrong@example.com), Password: wrong123 |
| Expected Output | 401 Unauthorized with error message |
| Status | Pass/Fail |

**✅ TC-003: Post New Project (Client)**

| **Test Case ID** | **TC-003** |
| --- | --- |
| Description | Client posts a project with valid data |
| Input | Title, Description, Budget |
| Expected Output | Project is saved and visible to freelancers |
| Status | Pass/Fail |

**✅ TC-004: Submit Proposal (Freelancer)**

| **Test Case ID** | **TC-004** |
| --- | --- |
| Description | Freelancer submits a proposal to an open project |
| Input | Project ID, Bid Amount, Message |
| Expected Output | Proposal is saved and visible to the client |
| Status | Pass/Fail |

**✅ TC-005: Access Admin Routes as Freelancer**

| **Test Case ID** | **TC-005** |
| --- | --- |
| Description | Try to call /api/admin/users as a freelancer |
| Expected Output | 403 Forbidden |
| Status | Pass/Fail |

**✅ TC-006: Payment Deposit (Client)**

| **Test Case ID** | **TC-006** |
| --- | --- |
| Description | Client deposits project amount via Razorpay |
| Input | Project ID, Payment Info |
| Expected Output | Razorpay redirects back with success status |
| Status | Pass/Fail |

**✅ TC-007: Chat Functionality**

| **Test Case ID** | **TC-007** |
| --- | --- |
| Description | Freelancer and client exchange real-time messages |
| Input | Message via WebSocket |
| Expected Output | Messages appear instantly in both users' chat views |
| Status | Pass/Fail |

**📊 5. Test Tools**

| **Tool** | **Purpose** |
| --- | --- |
| **JUnit/TestNG** | Unit Testing (Java) |
| **Postman** | Manual API Testing |
| **Selenium** | UI Automation (React pages) |
| **Mockito** | Service Layer Mocking |
| **JMeter** | Load/Performance Testing |
| **OWASP ZAP** | Security Testing |

**✅ 6. Acceptance Criteria**

| **Feature** | **Acceptance Criteria** |
| --- | --- |
| Registration/Login | Successful signup/login with JWT returned |
| Project Posting | Projects visible to freelancers instantly |
| Proposal Submission | Proposals viewable by respective clients |
| Messaging | Real-time, no delays, messages stored |
| Payments | Secure transactions, no duplicate charges |
| Admin Features | Only accessible to admin users |

**🚦 7. Bug Reporting Template**

| **ID** | **Summary** | **Steps to Reproduce** | **Expected** | **Actual** | **Priority** | **Status** |
| --- | --- | --- | --- | --- | --- | --- |
| 001 | Chat not loading | Login > Chat > Blank | Chat opens | Blank page | High | Open |

Would you like to proceed with:  
➡️ **Deployment Plan** – including environments, CI/CD flow, hosting, and versioning strategy?

If yes, reply **continue**.