|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Authentication** | |  |
|  | **Test Case** | **Steps** | **Expected Result** |
| 1 | Register with already used email | Enter email that is already registered, valid password and confirm password | Error message: "Email already in use" |
|
| 2 | Login with | Enter unregistered email and any password, click "Sign in" | Error message: "Invalid credentials / User not found" |
| unregistered email |
| 3 | Password length validation | Enter password shorter than required | Error message: "Password must be at least 8 characters long" |
| ( <8 characters), then click "Sign up" |
|  |
| 4 | Email field left blank | Leave email empty, enter passwords, click "Sign up" | Error message: "Email is required" |
|
| 5 | Password field left blank | Fill in email, leave password blank, | Error message: "Password is required" |
| click "Sign up" |
| 6 | Access protected page without login | Open user dashboard/profile URL without logging in | Redirected to login page |
|
| 7 | Case sensitivity in login | Register with certain case in email | Email may be case-insensitive; |
|  |
| 8 | Case sensitivity in password | Register with certain case in password | password must be case-sensitive |
| 9 | - | - |  |
|  |  |  |  |
| **Profile Management** | |  |  |
|  | **Test Case** | **Steps** | **Expected Result** |
| 10 | Invalid input in profile | Leave fields blank or enter special characters | Field-level validation or error messages |
|
| 11 | Remove URL | Click “Remove” next to a URL field | Field is removed from form |
| 12 | Save with required fields empty | Delete required info like username or email, click "Save" | Error message: "Required field |
| cannot be empty" |
| 13 | Upload profile picture | Click "Upload", select a valid image file | Image is uploaded, preview appears |
| 14 | Upload invalid profile picture | Upload non-image file or too large image | Error message: "Invalid file type or size" |
| 15 | Cancel profile edits | Make changes, then click “Cancel” | Form resets, original info remains |
| 16 | Auto-save feature (if applicable) | Change a field and wait without clicking “Save” | Changes saved automatically (if feature exists) |
| 17 | Accessibility check | Try navigating profile form using keyboard/tab key only | All fields/buttons are accessible by keyboard |
|  |  |  |  |
|  |  |  |  |
| **Tasks** | |  |  |
|  | **Test Case** | **Steps** | **Expected Result** |
| 18 | Task input validation | Enter invalid characters or formats (letters in budget, past deadline) | Field-level validation errors shown |
|
| 19 | Task list pagination | Create many tasks, navigate to next/prev page |  |
| Tasks are correctly paginated |
| 20 | Task sorting | Click to sort tasks by deadline or budget |  |
| Task list updates accordingly |
|  |
|  |
| 21 | Task filtering/search | Enter keyword or filter by date/status |  |
| Only relevant tasks are displayed |
| 22 | Attachments to task (if supported) | Upload a file during task creation/edit | File is saved and linked to the task |
| 23 | Deadline reminder feature (if supported) | Create task with upcoming deadline, wait or simulate time | Notification or reminder is triggered |
|
| 24 | Task visibility based on roles | Login as different role (e.g., admin/user), view tasks | Only permitted tasks are visible |
|
| 25 | Cancel task creation | Start creating a task, then click "Cancel" | Returned to task list, nothing saved |
| 26 | UI responsiveness | Resize window or use mobile view | Task form and list adapt to screen size |
|  |  |  |  |
| **Proposals** | |  |  |
|  | **Test Case** | **Steps** | **Expected Result** |
| 27 | Duplicate proposal prevention | Try to submit multiple proposals for same task | Error shown or previous proposal is updated |
|
| 28 | Edit proposal before acceptance | Click "Edit" on submitted proposal, update fields | Proposal updates are saved successfully |
|
| 29 | Proposal status updates in real-time | Freelancer refreshes page after client action | Proposal status reflects accepted/rejected/withdrawn |
|
| 30 | Unauthorized proposal actions | Another freelancer tries to withdraw/edit someone else's proposal | Access denied, error message shown |
|
|  |  | Submit a proposal, then check details | Timestamp of submission is visible |
| 31 |  |
|  | View proposal timestamp |
|  |  |
| 32 | Client cannot submit proposal to own task | As a task owner, try to submit a proposal | Button/option disabled or error shown |
|
| 33 | Notifications for new proposals (Client) | Freelancer submits a proposal | Task owner receives notification/email (if implemented) |
|
| 34 | Filter proposals by status | On "My Proposals" or task page, use filter | Only proposals matching selected status are shown |
|
| 35 | Proposal sorting | Sort proposals by budget, date, etc. | Sorted list is displayed correctly |
|
| 36 | Access control for accepted proposals |  | Access to relevant task features unlocked |
| After proposal is accepted, freelancer sees task and messaging |
|  |  |  |  |
|  |  |  |  |
| **Chat** | |  |  |
|  | **Test Case** | **Steps** | **Expected Result** |
| 37 | Typing indicator | Start typing in a chat window | Recipient sees "User is typing..." indicator |
| 38 | Send long message | Paste/send a very long message | Message is properly displayed/wrapped |
| 39 | Send special characters/emojis | Type emojis or special characters | Proper display without encoding issues |
| 40 | Chat timestamp visibility | Send/receive a message | Each message has a visible timestamp |
| 41 | Real-time sync on multiple devices | Send/receive message while logged in on multiple devices | All devices sync instantly |
| 42 | Message delivery/read status | Send a message, wait for recipient to read | "Delivered" or "Seen" indicators appear (if supported) |
| 43 | Block user | Block another user, try messaging them | Chat disabled, block persists across sessions |
| 44 | Chat notification | Receive message while not in chat | Notification icon or sound alert shown |
|
| 45 | Report abusive message | Click "Report" on a specific message | Message is flagged, feedback sent to moderation (if implemented) |
| 46 | Chat search | Use search bar to find a message in chat history | Relevant messages are highlighted or listed |
| 47 | Scroll up for older messages | Scroll up in chat window | Older messages load via lazy loading or pagination |
| 48 | Chat session timeout | Leave chat open for a long time, wait | Session auto-refreshes or stays connected without error |
| 49 | File size limit | Try uploading a file exceeding max size | Error message shown: “File too large” |
|
| 50 | Unsupported file type | Try uploading an unsupported format (e.g., .exe) | Upload blocked, error message shown |
|  |  |  |  |
|  |  |  |  |
| **Performance** | |  |  |
|  | **Test Case** | **Steps** | **Expected Result** |
| 51 | Stress test login endpoint | Simulate 1000+ login requests | No timeout or crash, proper throttling |
|
| 52 | Profile update | Simulate 100 users updating profiles simultaneously | All updates processed correctly |
| under load |
| 53 | Chat flood test | Send 100+ messages | App remains stable, messages delivered in order |
| in under 1 minute |
| 54 | Proposal spam protection | Attempt to send proposals rapidly using automation | Rate limiting or CAPTCHA triggered |
|
|  |  |  |  |
| **Security** | |  |  |
|  | **Test Case** | **Steps** | **Expected Result** |
| 55 | Session expiration | Leave session idle for a long time | User is logged out automatically |
| (10 min) |
| 56 | CSRF(Cross Site | Submit unauthorized form from external site | Request is blocked or denied |
| Request Forgery) |
| protection |
|  |
| 57 | File upload validation | Upload executable or renamed file as image | Upload is blocked or sanitized |
| 58 | Unauthorized API access | Try accessing protected endpoints without login | 401 Unauthorized or redirect to login |
| 59 | Brute force login protection | Attempt multiple failed logins | Account lock or CAPTCHA triggered |
| 60 | Role-based access | Login as different roles (freelancer/client/admin) | Only permitted actions are allowed |