

# QA Intern Assignment - User Authentication & Authorization (JWT)

## Manual Test Case - User Registration

**Test Case ID:**TC\_REG\_01

**Title:** Verify successful user registration with valid details (valid name, email, password, confirm password).

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1.	Navigate to Registration page	TD_01 (Name: Ashwini K B, Email:	<b>UI:</b> Registration form should be displayed with all required fields.
2.	Enter valid Name, Email, Password, Confirm Password	kbashwini450@gmail.com, Password: ashukb@1234>, Confirm Password:	<b>UI:</b> All Fields should be accepted (no validation errors)
3.	Click Register button	ashukb@1234>)	<b>UI:</b> Success message should be displayed, user redirected to Login page <b>API/Postman:</b> 201 Created should be returned, JWT token should be generated in response.

**Priority:** High.

**Test Case ID:**TC\_REG\_02

**Title:** Verify successful user registration with valid details (Name, Email, Password, Confirm Password)

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Test data **TD\_02 (anu200@gmail.com)** is not previously registered.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1.	Navigate to Registration page	TD_02 (Name: ASHWINI.K.B, Email:	<b>UI:</b> Registration form should be displayed with all required fields.
2.	Enter valid Name, Email, Password, Confirm Password	anu200@gmail.com, Password: kabsce#\$6734, Confirm Password:	<b>UI:</b> All Fields should be accepted (no validation errors)
3.	Click Register button	kabsce#\$6734)	<b>UI:</b> Success message should be displayed, user redirected to Login page

			<b>API/Postman:</b> 201 Created should be returned, JWT token should be generated in response
--	--	--	---

**Priority:** High.

**Test Case ID:**TC\_REG\_03

**Title:** Verify user registration with uppercase email and minimum length name.

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Test data **TD\_03 (ANU200@GMAIL.COM)** is not previously registered.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Registration page	TD_03 (Name: deepu, Email: ANU200@GMAIL.COM, Password: abc123456789, Confirm Password: abc123456789)	<b>UI:</b> Registration form should be displayed with all fields empty and ready for input.
2	Enter valid Name, valid Email in Uppercase, valid Password, Confirm Password		<b>UI:</b> All Fields should be accepting input (no validation errors). Name is <b>exactly 5 characters</b> , meets minimum length. Email should be accepted without validation error
3	Click Register button		<b>UI:</b> Success message should be displayed, user redirected to Login page. <b>API/Postman:</b> 201 Created should be returned; email should be processed according to requirement (converted to lowercase); JWT token should be generated.

**Priority:** High.

**Test Case ID:**TC\_REG\_04

**Title:** Verify user registration with numeric-only name that meets minimum length requirement.

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Test data **TD\_04** is not previously registered.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Registration page	TD_04 (Name: 12345, Email:	<b>UI:</b> Registration form should be displayed with all required fields.

2	Enter numeric-only Name, valid Email, valid Password and Confirm Password	surajkali23@gmail.com, Password: ABCdef123456, Confirm Password: ABCdef123456)	<b>UI:</b> Name field should be accepted numeric input since only length validation (5–24) is defined. No validation error shown.
3	Click on Register button		<b>UI:</b> Registration successful message should be displayed; user should be redirected to Login page. <b>API: 201 Created</b> and success message should be returned.

**Priority:** Medium.

**Test Case ID:**TC\_REG\_05

**Title:** Verify user registration fails when using an already registered (duplicate) email.

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Email **anu200@gmail.com** already exists in the system.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Registration page	TD_05 (Name: ashu@123, Email: anu200@gmail.com, Password: Ashwini@1234, Confirm Password: Ashwini@1234)	<b>UI:</b> Registration form should be displayed with all required fields.
2	Enter valid Name, Existing Email, valid Password and Confirm Password		<b>UI:</b> All fields should be accept input
3	Click on Register button		<b>UI:</b> Error message should be displayed (e.g., “Email already registered”). User stays on Registration page. <b>API: 409 Conflict</b> should be (Duplicate email) returned

**Priority:** High.

**Test Case ID:**TC\_REG\_06

**Title:** Verify user registration fails when password contains only letters and no numeric characters.

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Test data **TD\_06** is not previously registered.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
-------	--------	-----------	-----------------

1	Navigate to Registration page	TD_06 (Name: ,,,",,,,,,.....""""", Email: kavyathanu34589007545353wqrrt	<b>UI:</b> Registration form should be displayed with all required fields.
2	Enter Name, valid Email, Password containing only letters, and Confirm Password	@gmail.com, Password: OnlyLetters, Confirm Password: OnlyLetters)	<b>UI:</b> Password validation error should be shown indicating password must contain letters and numbers
3	Click on Register button		<b>UI:</b> Registration should be blocked. <b>API:</b> 400 Bad Request should be returned

**Priority:** High.

**Test Case ID:**TC\_REG\_07

**Title:** Verify user registration fails when email format contains multiple “@” symbols.

### Preconditions:

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Test data **TD\_07** is not previously registered.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Registration page	TD_07 (Name: Anita Singh, Email:	<b>UI:</b> Registration form should be displayed with all required fields.
2	Enter valid Name, invalid Email format, valid Password and Confirm Password	user@@gmail.com, Password: StrongPassword12, Confirm Password:	<b>UI:</b> Email validation error should be displayed
3	Click on Register button	StrongPassword12)	<b>UI:</b> Registration should be blocked. <b>API:</b> 400 Bad Request should be returned

**Priority:** High.

**Test Case ID:**TC REG 08

**Title:** Verify user registration fails when Confirm Password does not match Password.

### Preconditions:

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Test data **TD\_08** is not previously registered.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Registration page	TD_08 (Name: SanaKB, Email:	<b>UI:</b> Registration form should be displayed with all required fields.

2	Enter valid Name, Email, Password, and mismatched Confirm Password	user@gmail.com, Password:	<b>UI:</b> Confirm Password validation error should be displayed
3	Click on Register button	testpassword12, Confirm Password: StrongPassword12)	<b>UI:</b> Registration should be blocked. <b>API:</b> 400 Bad Request should be returned

**Priority:** High.

**Test Case ID:**TC\_REG\_09

**Title:** Verify user registration fails when password contains only numeric characters.

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Test data **TD\_09** is not previously registered.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Registration page	TD_09 (Name: samirak, Email:	<b>UI:</b> Registration form should be displayed with all required fields.
2	Enter Name, valid Email, numeric-only Password, and Confirm Password	ashumk345@gmail.com, Password: 98765432109, Confirm Password: 98765432109)	<b>UI:</b> Password validation error should be displayed indicating password must contain letters and numbers
3	Click on Register button		<b>UI:</b> Registration should be blocked. <b>API:</b> should return 400 Bad Request

**Priority:** High.

**Test Case ID:**TC\_REG\_010

**Title:** Verify user registration fails when name length is below the minimum required characters.

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Test data **TD\_10** should not be previously registered.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Registration page	TD_10 (Name: 12, Email:	<b>UI:</b> Registration form should be displayed with all required fields.
2	Enter Name with less than 5 characters and valid Email, Password, Confirm Password	kiran450@gmail.com, Password: pass12word345, Confirm Password: pass12word345)	<b>UI:</b> Name validation error should be displayed indicating minimum length requirement.
3	Click on Register button		<b>UI:</b> Registration should be blocked.

			<b>API:</b> should return 400 Bad Request
--	--	--	---

**Priority:** High.

**Test Case ID:**TC\_REG\_011

**Title:** Verify user registration behaviour when name is provided in numeric/scientific notation format.

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Test data **TD\_11** should not be previously registered.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Registration page	TD_11 (Name: 1.23457E+24, Email: sanamk@gmail.com, Password: Pass@1234567, Confirm Password: Pass@1234567)	<b>UI:</b> Registration form should be displayed with all required fields.
2	Enter Name in numeric/scientific notation format and valid Email, Password, Confirm Password.		<b>UI:</b> Name field input should be accepted as length rule is satisfied.
3	Click on Register button		<b>UI:</b> Registration should be successful or rejected based on backend validation rules. <b>API</b> should return 201 Created or 400 Bad Request accordingly

**Priority:** Medium.

**Test Case ID:**TC\_REG\_012

**Title:** Verify user registration with name containing mixed case letters, numbers, spaces and special characters.

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Test data **TD\_12** should not be previously registered.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Registration page	TD_10 (Name: AsHu1,ASHU12<:'])]@#\$\$%&*, Email: <a href="mailto:deepu34@gmail.com">deepu34@gmail.com</a> , Password: TestPassword, Confirm Password: TestPassword)	<b>UI:</b> Registration form should be displayed with all required fields.
2	Enter Name with mixed characters and valid Email, Password and Confirm Password		<b>UI:</b> Name field should accept input since no character restriction is specified

3	Click on Register button		<b>UI:</b> Registration should be successful. <b>API</b> should return 201 Created
---	--------------------------	--	--

**Priority:** Medium

**Test Case ID:**TC\_REG\_013

**Title:** Verify user registration fails with invalid email format.

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Test data **TD\_13** should not be previously registered.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Registration page	TD_13 (Name: 12345, Email:	<b>UI:</b> Registration form should be displayed with all required fields.
2	Enter Name, invalid Email format, valid Password and Confirm Password	anjanak47@.com, Password: abc123456789, Confirm Password:	<b>UI:</b> Email validation error should be displayed
3	Click on Register button	abc123456789)	<b>UI:</b> Registration should be blocked. <b>API</b> should return 400 Bad Request

**Priority:** High.

**Test Case ID:**TC\_REG\_014

**Title:** Verify user registration fails when password length is below the minimum required.

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Test data **TD\_14** should not be previously registered.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Registration page.	TD_14 (Name: ashuk, Email:	<b>UI:</b> Registration form should be displayed with all required fields.
2	Enter Name, valid Email, Password with less than 12 characters, and Confirm Password.	barsha@gmail.com, Password: loginPass12, Confirm Password: loginPass12)	<b>UI:</b> Password validation error should be displayed indicating minimum 12 characters are required.
3	Click on Register button.		<b>UI:</b> Registration should be blocked. <b>API</b> should return 400 Bad Request.

**Priority:** High.

**Test Case ID:**TC\_REG\_015

**Title:** Verify user registration with very long but valid email address.

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Test data **TD\_15** should not be previously registered.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Registration page	TD_15 (Name: barsha, Email: very long email, Password: userName123, Confirm Password: userName123)	<b>UI:</b> Registration form should be displayed with all required fields
2	Enter Name, very long valid Email, Password and Confirm Password		<b>UI:</b> Email field should accept input if format is valid
3	Click on Register button		<b>UI:</b> Registration should be successful. API should return 201 Created

**Priority:** Medium

## Manual Test Case – User Login

**Test Case ID:**TC\_REG\_016

**Title:** Verify login with valid email and password

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- User from Test data **TD\_01** is already registered successfully
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Login page		<b>UI:</b> Login page should be displayed with all required fields.
2	Enter invalid Email	TD_01.Email (kbashwini450@gmail.com)	<b>UI:</b> Email field should accept input
3	Enter valid Password	TD_01. Password ( <a href="#">ashukb@1234&gt;</a> )	<b>UI:</b> Password field should accept input
4	Click Login button		<b>UI:</b> User should be redirected to Dashboard, <b>API</b> should return JWT token.

**Priority:** High



**Test Case ID:**TC\_REG\_017

**Title:** Verify login with valid email and invalid password.

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- User from Test data **TD\_01** is already registered successfully
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Login page		<b>UI:</b> Login page should be displayed with all required fields.
2	Enter invalid Email	TD_01.Email (kbashwini450@gmail.com)	<b>UI:</b> Email field should accept input
3	Enter valid Password	WrongPass123	<b>UI:</b> Password field should accept input
4	Click Login button		<b>UI:</b> error message should be displayed Wrong password, <b>API</b> should return <b>401 Unauthorized</b> .

**Priority:** High

**Test Case ID:**TC\_REG\_018

**Title:** Verify login with unregistered email

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- User from Test data **TD\_10** unregistered.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Login page		<b>UI:</b> Login page should be displayed with all required fields.
2	Enter invalid Email	TD_0.Email (kiran450@gmail.com)	<b>UI:</b> Email field should accept input
3	Enter valid Password	TD_01. Password ( pass12word345)	<b>UI:</b> Password field should accept input
4	Click Login button		<b>UI:</b> error message should be displayed invalid Email or password, <b>API</b> should return <b>401 Unauthorized</b>

**Priority:** High

**Test Case ID:**TC\_REG\_019

**Title:** Verify UI validation when Email and Password fields are left empty during login.

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Login page		<b>UI:</b> Login page should be displayed with Email and Password fields
2	Leave Email and Password empty.		<b>UI:</b> Validation error messages should be displayed for required fields
3	Click Login button		<b>UI:</b> Login action should be blocked; <b>API:</b> Login request should not be triggered

**Priority:** High

**Test Case ID:**TC\_REG\_020

**Title:** Verify login is blocked when Email field contains invalid special characters

**Preconditions:**

- Web application is up and accessible.
- User is not logged in.
- Home page is open.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Login page		<b>UI:</b> Login page should be displayed with Email and Password fields
2	Leave Email and Password empty.	Email: anu@##@gmail.com	<b>UI:</b> Email validation error message should be displayed.
3	Click Login button		<b>UI:</b> Login action should be blocked; <b>API:</b> Login request should not be triggered

**Priority:** Medium.

**Manual Test Case - Dashboard (Protected Page)**

**Test Case ID:**TC\_REG\_021

**Title:** Verify dashboard access for logged-in user with valid JWT

**Preconditions:**

- Web application is up and accessible.
- User is successfully logged in using valid credentials
- Valid JWT token is generated and stored
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Login with valid email and password	TD_01.Email (kbashwini450@gmail.com) TD_01.Password (ashukb@1234>*)	<b>API:</b> Should return JWT token; <b>UI:</b> User should be successfully logged in
2	Navigate to Dashboard		<b>UI:</b> Dashboard should be displayed as the user is authenticated with a valid JWT
3	Verify logged-in user details		<b>UI:</b> Logged-in user email and route name should be displayed

**Priority:** High.

**Test Case ID:**TC\_REG\_022

**Title:** Verify user is redirected to Login page when accessing Dashboard without logging in.

**Preconditions:**

- Web application is up and accessible.
- User is not registered or not logged in
- No JWT token is present in browser storage
- Backend API and database are running

Steps	Action	Test Data	Expected Result
1	Open Home page		<b>UI:</b> Home page should be displayed
2	Attempt to access Dashboard URL directly	URL: /dashboard	Backend: Should validate JWT token
3	Observe system behaviour		<b>API:</b> Should return 401 Unauthorized; <b>UI:</b> User should be redirected to Login page

**Priority:** High.

**Test Case ID:**TC\_REG\_023

**Title:** Verify dashboard displays logged-in user email and route name.

**Preconditions:**

- Web application is up and accessible.
- User is successfully logged in using valid credentials
- Valid JWT token is generated and stored
- Valid JWT token exists.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Dashboard		<b>UI:</b> Dashboard should be displayed
2	Verify displayed user details		<b>UI:</b> Logged-in user email and route name should be displayed correctly

**Priority:** High.

**Test Case ID:**TC\_REG\_024

**Title:** Verify JWT token is invalid after logout.

**Preconditions:**

- Web application is up and accessible.
- User is successfully logged in using valid credentials
- Valid JWT token is generated and stored
- Valid JWT token exists.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Login with valid email and password	TD_01.Email (kbashwini450@gmail.com) TD_01.Password (ashukb@1234>*)	<b>API:</b> Should return JWT token; <b>UI:</b> User should be successfully logged in
2	Navigate to Dashboard		<b>UI:</b> Dashboard should be displayed as the user is authenticated with a valid JWT
3	Verify logged-in user details		<b>UI:</b> Logged-in user email and route name should be displayed
4	Logout from application		<b>UI:</b> User should be redirected to Login page
5	UI: User should be redirected to Login page		<b>UI:</b> User should be redirected to Login page

**Priority:** High.

**Manual Test Case - Rate Limiting.**

**Test Case ID:**TC\_REG\_025

**Title:** Verify rate limiting after multiple invalid login attempts.

**Preconditions:**

- Web application is up and accessible.
- Home page is open.
- User is not logged in.
- Same IP address is used.
- Backend API and database are running.

Steps	Action	Test Data	Expected Result
1	Navigate to Login page		Login page should be displayed
2	Enter invalid Email	Email: notregistered@gmail.com	Email field should accept input
3	Enter invalid Password	Password: WrongPass123	Password field should accept input
4	Click Login button		Error message should be displayed: “Invalid credentials”
5	Repeat steps 2–4 ten times	Same invalid credentials	<b>API</b> should return <b>401 Unauthorized</b>
6	Click Login button for the 11th attempt	Same invalid credentials	<b>API</b> should return <b>429 Too Many Requests</b> and UI should show rate-limit message

**Priority:** High.