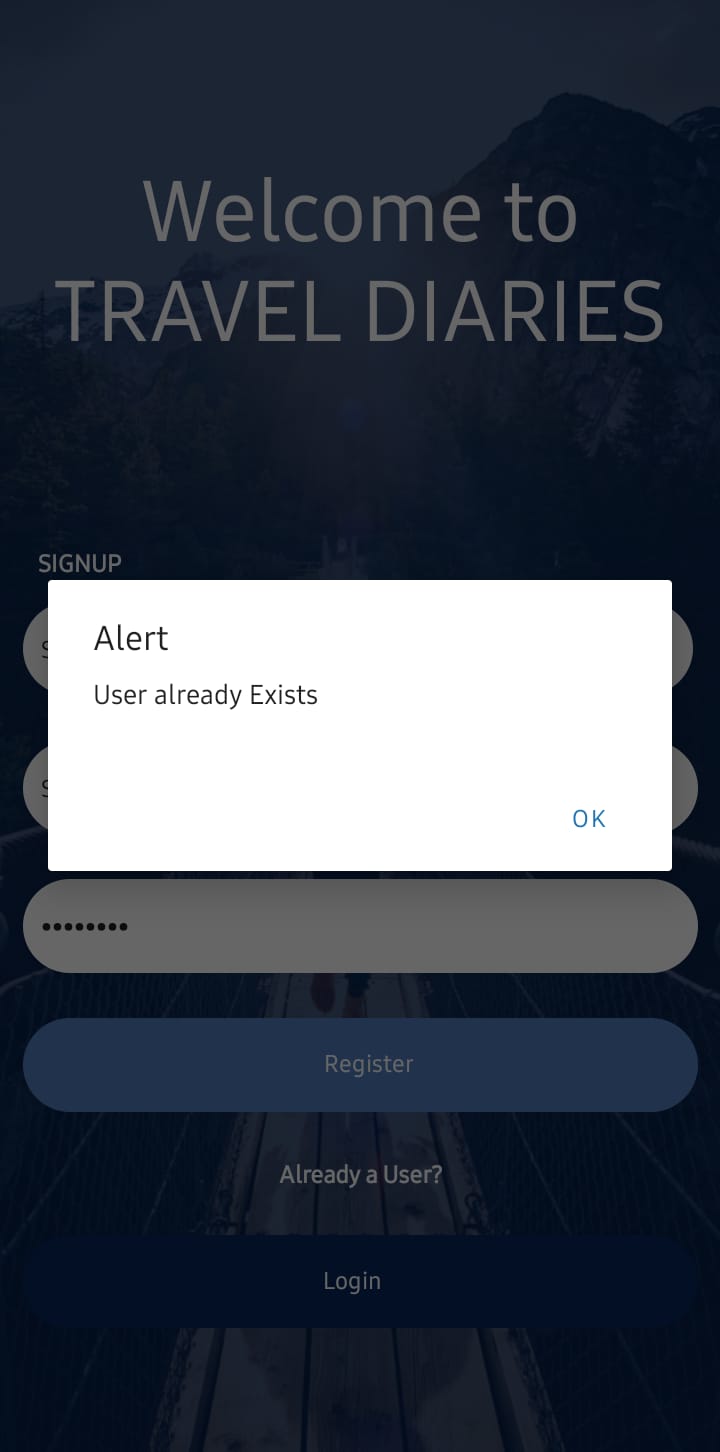
|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 1 | **Project Name** | Travel Diary |
| **Test Case Id** | 1.1 | **Test Type** | Functionality |
| **Test Case Description** | Verify Splash Screen | | |
| **Test Steps** | Open the app | | |
| **Expected Result** | Splash screen is displayed. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |



|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 1 | **Project Name** | Travel Diary |
| **Test Case Id** | 1.2 | **Test Type** | Functionality |
| **Test Case Description** | Verify that users can successfully create accounts. | | |
| **Test Steps** | 1. Open the app. 2. Navigate to the account creation page. 3. Enter valid user details. 4. Submit the form. | | |
| **Expected Result** | The user account is created successfully. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **­Tested By** | Bushra Ahmed | | |

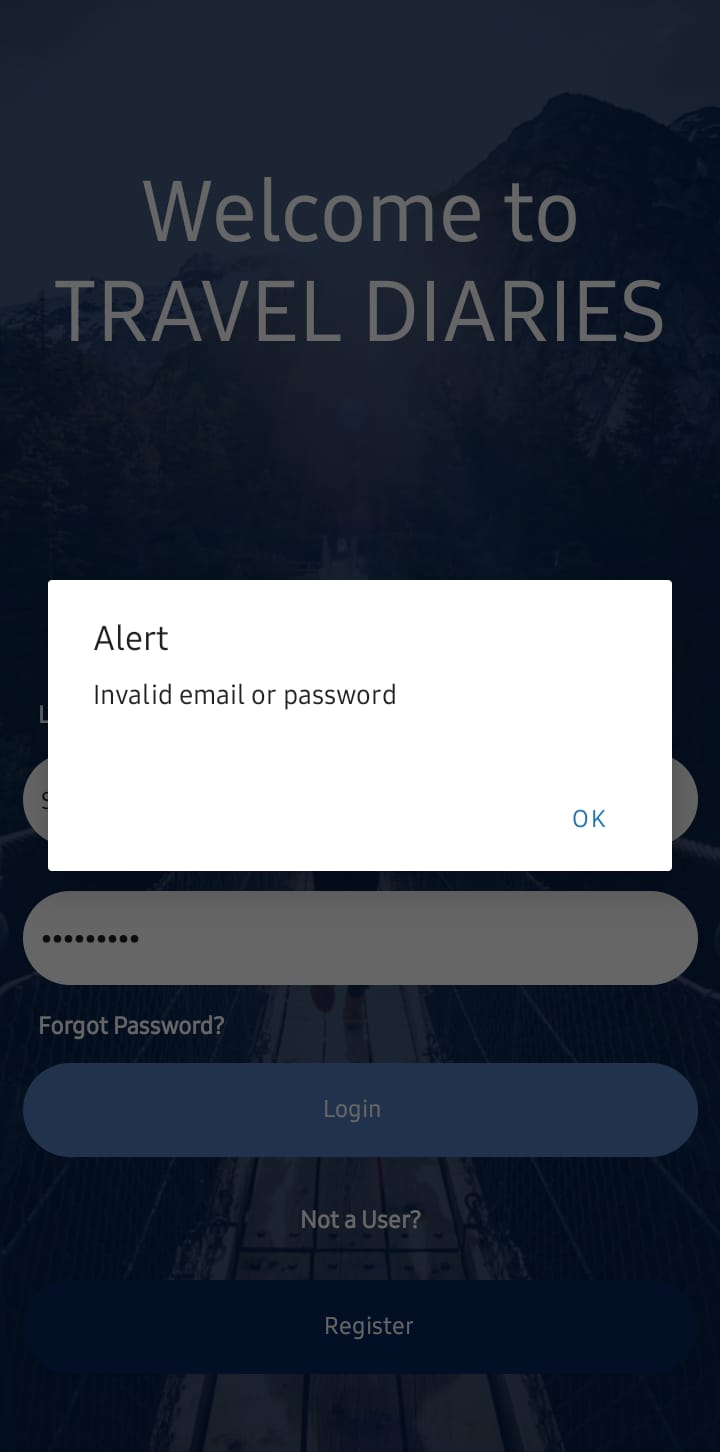
|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 1 | **Project Name** | Travel Diary |
| **Test Case Id** | 1.3 | **Test Type** | Functionality |
| **Test Case Description** | Duplicate Email Detection during Registration | | |
| **Test Steps** | 1. Open the app. 2. Navigate to the user registration or sign-up page 3. Enter valid user details. 4. Submit the form. | | |
| **Expected Result** | The system detects that the email address [ExistingUserEmail@example.com] is already registered. OR  Display error: "User with this email already exists. Please log in or use a different email." | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **­Tested By** | Bushra Ahmed | | |



|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 1 | **Project Name** | Travel Diary |
| **Test Case Id** | 1.3 | **Test Type** | Functionality |
| **Test Case Description** | Verify that users cannot create accounts with invalid or incomplete information. | | |
| **Test Steps** | 1. Open the app. 2. Navigate to the account creation page. 3. Provide incomplete or invalid details. 4. Attempt to submit the form. | | |
| **Expected Result** | Account creation fails, and appropriate error messages are displayed. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 1 | **Project Name** | Travel Diary |
| **Test Case Id** | 1.4 | **Test Type** | Functionality |
| **Test Case Description** | Ensure users can securely log in. | | |
| **Test Steps** | 1. Open the app. 2. Navigate to the login page. 3. Enter valid login credentials. 4. Log in. | | |
| **Expected Result** | The user is securely logged in. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 1 | **Project Name** | Travel Diary |
| **Test Case Id** | 1.5 | **Test Type** | Functionality |
| **Test Case Description** | Ensure users cannot log in with incorrect credentials. | | |
| **Test Steps** | 1. Open the app. 2. Navigate to the login page. 3. Enter incorrect login credentials. 4. Attempt to log in. | | |
| **Expected Result** | Login fails, and users receive an authentication error. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |



|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 1 | **Project Name** | Travel Diary |
| **Test Case Id** | 1.6 | **Test Type** | Functionality |
| **Test Case Description** | Verify that users can log out. | | |
| **Test Steps** | 1. Open the app. 2. Log in with valid credentials. 3. Navigate to the logout option. | | |
| **Expected Result** | The user is successfully logged out. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 2 | **Project Name** | Travel Diary |
| **Test Case Id** | 2.1 | **Test Type** | Functionality |
| **Test Case Description** | Confirm users can create new journeys. | | |
| **Test Steps** | 1. Open the app. 2. Navigate to the journey creation section. 3. Provide a title, theme, and cover photo. 4. Save or submit the new journey. | | |
| **Expected Result** | A new journey is created and visible in the user's profile. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 2 | **Project Name** | Travel Diary |
| **Test Case Id** | 2.2 | **Test Type** | Functionality |
| **Test Case Description** | Confirm that users cannot create journeys with duplicate titles. | | |
| **Test Steps** | 1. Open the app. 2. Create a new journey with a title that already exists. | | |
| **Expected Result** | The app should reject the creation of a journey with a duplicate title. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 3 | **Project Name** | Travel Diary |
| **Test Case Id** | 3.1 | **Test Type** | Functionality |
| **Test Case Description** | Ensure users can create visually appealing photo journals. | | |
| **Test Steps** | 1. Open the app. 2. Access the photo journal creation feature. 3. Add photos with captions. 4. Save or submit the photo journal. | | |
| **Expected Result** | A photo journal is created successfully. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 3 | **Project Name** | Travel Diary |
| **Test Case Id** | 3.2 | **Test Type** | Functionality |
| **Test Case Description** | Ensure users cannot upload files with unsupported formats. | | |
| **Test Steps** | 1. Open the app. 2. Attempt to upload a file in an unsupported format. | | |
| **Expected Result** | The app should reject unsupported file types and provide an error message. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 4 | **Project Name** | Travel Diary |
| **Test Case Id** | 4.1 | **Test Type** | Functionality |
| **Test Case Description** | Confirm that the user's journey is displayed on the map. | | |
| **Test Steps** | 1. Open a created journey. 2. Navigate to the map view. 3. Verify clickable locations are displayed. | | |
| **Expected Result** | The journey is accurately displayed on the interactive map. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 4 | **Project Name** | Travel Diary |
| **Test Case Id** | 4.2 | **Test Type** | Functionality |
| **Test Case Description** | Verify that users cannot click on locations that are not part of their journey. | | |
| **Test Steps** | 1. Open the app. 2. Access the map view for a journey. 3. Attempt to click on locations outside the current journey. | | |
| **Expected Result** | Clicking on unavailable locations should have no effect. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 5 | **Project Name** | Travel Diary |
| **Test Case Id** | 5.1 | **Test Type** | Functionality |
| **Test Case Description** | Verify users can share their travel experiences on social media. | | |
| **Test Steps** | 1. Open a created journey or photo journal. 2. Access the sharing option. 3. Share on a social media platform. | | |
| **Expected Result** | The content is shared on the selected social media platform. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 5 | **Project Name** | Travel Diary |
| **Test Case Id** | 5.2 | **Test Type** | Functionality |
| **Test Case Description** | Confirm that users cannot share content without proper authorization. | | |
| **Test Steps** | 1. Open the app. 2. Attempt to share content without proper permissions. | | |
| **Expected Result** | The app should prevent unauthorized sharing. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 6 | **Project Name** | Travel Diary |
| **Test Case Id** | 6.1 | **Test Type** | Functionality |
| **Test Case Description** | Confirm users have control over the privacy settings of their journeys. | | |
| **Test Steps** | 1. Open the app. 2. Navigate to account settings. 3. Adjust the privacy settings of a journey. | | |
| **Expected Result** | Privacy settings are applied as configured. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 6 | **Project Name** | Travel Diary |
| **Test Case Id** | 6.2 | **Test Type** | Functionality |
| **Test Case Description** | Verify that users cannot change the privacy settings of journeys without proper authorization. | | |
| **Test Steps** | 1. Open the app. 2. Navigate to the privacy settings of another user's journey. 3. Attempt to modify the privacy settings. | | |
| **Expected Result** | The app should prevent unauthorized changes to privacy settings. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 7 | **Project Name** | Travel Diary |
| **Test Case Id** | 7.1 | **Test Type** | Functionality |
| **Test Case Description** | Ensure users can explore journeys created by others. | | |
| **Test Steps** | 1. Open the app. 2. Navigate to the explore section. 3. Browse and select a journey created by another user. | | |
| **Expected Result** | The selected journey is displayed for exploration. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 7 | **Project Name** | Travel Diary |
| **Test Case Id** | 7.2 | **Test Type** | Functionality |
| **Test Case Description** | Confirm that users cannot perform searches with invalid or nonsensical keywords. | | |
| **Test Steps** | 1. Open the app. 2. Access the search feature. 3. Enter invalid or nonsensical keywords. | | |
| **Expected Result** | The app should handle and provide appropriate feedback for invalid search queries. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 7 | **Project Name** | Travel Diary |
| **Test Case Id** | 7.3 | **Test Type** | Functionality |
| **Test Case Description** | Verify users can search for specific locations or themes. | | |
| **Test Steps** | 1. Open the app. 2. Access the search feature. 3. Enter specific keywords, locations, or themes. | | |
| **Expected Result** | Relevant results are displayed based on the search criteria. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 7 | **Project Name** | Travel Diary |
| **Test Case Id** | 7.4 | **Test Type** | Functionality |
| **Test Case Description** | Ensure that users cannot explore or access unpublished or private journeys without proper authorization. | | |
| **Test Steps** | 1. Open the app. 2. Attempt to explore private or unpublished journeys. | | |
| **Expected Result** | The app should prevent unauthorized access to private journeys. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 7 | **Project Name** | Travel Diary |
| **Test Case Id** | 7.5 | **Test Type** | Functionality |
| **Test Case Description** | Confirm users can discover diverse travel experiences. | | |
| **Test Steps** | 1. Open the app. 2. Explore journeys from different users. 3. Note the variety in themes, locations, and experiences. | | |
| **Expected Result** | Users can discover a diverse range of travel experiences. | | |
| **Actual Result** |  | | |
| **Pass/Fail** | Pass | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 25/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 8 | **Project Name** | Travel Diary |
| **Test Case Id** | 8.1 | **Test Type** | Backend |
| **Test Case Description** | Successful user registration. | | |
| **Test Steps** | 1. Send a POST request to register a user with valid details. 2. Verify the user is inserted into the SQL Server database. 3. Check that the API returns a 200 Created status. | | |
| **Expected Result** | Successful registration with a new user in the database, and API responds with a 200 status. | | |
| **Actual Result** |  | | |
| **Pass/Fail** |  | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 26/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 8 | **Project Name** | Travel Diary |
| **Test Case Id** | 8.2 | **Test Type** | Backend |
| **Test Case Description** | Duplicate user registration prevention. | | |
| **Test Steps** | 1. Attempt to register a user with the same credentials. 2. Verify the API responds with a 400 Conflict status. 3. Check that the user count in the database remains unchanged. | | |
| **Expected Result** | API prevents duplicate registrations with a 400 status, and user count remains unchanged. | | |
| **Actual Result** |  | | |
| **Pass/Fail** |  | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 26/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 8 | **Project Name** | Travel Diary |
| **Test Case Id** | 8.3 | **Test Type** | Backend |
| **Test Case Description** | Successful user login. | | |
| **Test Steps** | Send a POST request to login with valid credentials.  Confirm the API returns a valid authentication token.  Check that the token is required for subsequent authenticated requests. | | |
| **Expected Result** | Successful login with a valid token required for subsequent requests. | | |
| **Actual Result** |  | | |
| **Pass/Fail** |  | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 26/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 8 | Project Name | Travel Diary |
| **Test Case Id** | 8.4 | Test Type | Backend |
| **Test Case Description** | Invalid user login prevention. | | |
| **Test Steps** | Attempt to log in with incorrect credentials.  Verify the API responds with a 400 Unauthorized status. | | |
| **Expected Result** | API prevents invalid logins with a 400 status | | |
| **Actual Result** |  | | |
| **Pass/Fail** |  | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 26/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 9 | **Project Name** | Travel Diary |
| **Test Case Id** | 9.1 | **Test Type** | Backend |
| **Test Case Description** | Send a POST request with valid diary entry data and a valid authentication token. | | |
| **Test Steps** | 1. Confirm the entry is correctly inserted into the SQL Server database. 2. Check that the API returns a 200 Created status. | | |
| **Expected Result** | Successful creation of a diary entry in the database, and API responds with a 200 status. | | |
| **Actual Result** |  | | |
| **Pass/Fail** |  | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 26/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 9 | **Project Name** | Travel Diary |
| **Test Case Id** | 9.2 | **Test Type** | Backend |
| **Test Case Description** | Update an existing diary entry. | | |
| **Test Steps** | 1. Send a PUT/PATCH request to update an entry with a valid authentication token. 2. Confirm the entry in the database is correctly updated. 3. Check that the API returns a 200 OK status. | | |
| **Expected Result** | Successful update of a diary entry in the database, and API responds with a 200 status. | | |
| **Actual Result** |  | | |
| **Pass/Fail** |  | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 26/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 9 | **Project Name** | Travel Diary |
| **Test Case Id** | 9.3 | **Test Type** | Backend |
| **Test Case Description** | Diary entry insertion with invalid authentication. | | |
| **Test Steps** | 1. Attempt to create a new entry without a valid authentication token. 2. Verify the API responds with a 400 Unauthorized status. 3. Confirm the entry is not inserted into the database. | | |
| **Expected Result** | API prevents entry creation without valid authentication with a 400 status, and entry is not added. | | |
| **Actual Result** |  | | |
| **Pass/Fail** |  | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 26/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 9 | **Project Name** | Travel Diary |
| **Test Case Id** | 9.4 | **Test Type** | Backend |
| **Test Case Description** | Diary entry deletion. | | |
| **Test Steps** | 1. Send a DELETE request to delete an entry with a valid authentication token. 2. Confirm the entry is removed from the SQL Server database. 3. Check that the API returns a 200 No Content status. | | |
| **Expected Result** | Successful deletion of a diary entry in the database, and API responds with a 200 status. | | |
| **Actual Result** |  | | |
| **Pass/Fail** |  | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 26/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 9 | **Project Name** | Travel Diary |
| **Test Case Id** | 9.1 | **Test Type** | Backend |
| **Test Case Description** | Send a POST request with valid diary entry data and a valid authentication token. | | |
| **Test Steps** | 1. Confirm the entry is correctly inserted into the SQL Server database. 2. Check that the API returns a 200 Created status. | | |
| **Expected Result** | Successful creation of a diary entry in the database, and API responds with a 200 status. | | |
| **Actual Result** |  | | |
| **Pass/Fail** |  | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 26/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 10 | **Project Name** | Travel Diary |
| **Test Case Id** | 10.1 | **Test Type** | Backend |
| **Test Case Description** | Successful user logout. | | |
| **Test Steps** | 1. Send a POST request to logout with a valid authentication token. 2. Confirm the token is invalidated and no longer usable. 3. Check that the API returns a 200 OK status. | | |
| **Expected Result** | Successful logout with the invalidated token and API responds with a 200 status. | | |
| **Actual Result** |  | | |
| **Pass/Fail** |  | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 26/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 10 | **Project Name** | Travel Diary |
| **Test Case Id** | 10.2 | **Test Type** | Backend |
| **Test Case Description** | Attempt to access protected resources after logout. | | |
| **Test Steps** | 1. Try to perform authenticated operations after logout using the invalidated token. 2. Verify the API responds with a 400 Unauthorized status. | | |
| **Expected Result** | API prevents access to protected resources after logout with a 400 status. | | |
| **Actual Result** |  | | |
| **Pass/Fail** |  | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 26/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 11 | **Project Name** | Travel Diary |
| **Test Case Id** | 11.1 | **Test Type** | Backend |
| **Test Case Description** | Verify data integrity after user registration. | | |
| **Test Steps** | 1. Check that the user data is correctly inserted into the SQL Server database. 2. Validate that the user information matches the provided registration details. | | |
| **Expected Result** | User data is accurately inserted, and information aligns with registration details. | | |
| **Actual Result** |  | | |
| **Pass/Fail** |  | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 26/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 11 | **Project Name** | Travel Diary |
| **Test Case Id** | 11.2 | **Test Type** | Backend |
| **Test Case Description** | Verify data integrity after diary entry operations. | | |
| **Test Steps** | 1. Confirm diary entries are correctly inserted, updated, and deleted in the SQL Server database. 2. Validate that the database state reflects the expected changes. | | |
| **Expected Result** | Diary entry operations accurately reflect in the SQL Server database. | | |
| **Actual Result** |  | | |
| **Pass/Fail** |  | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 26/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 12 | **Project Name** | Travel Diary |
| **Test Case Id** | 12.1 | **Test Type** | Backend |
| **Test Case Description** | Test for SQL Server connection issues. | | |
| **Test Steps** | 1. Introduce temporary connectivity issues with the SQL Server. 2. Ensure the API handles the issue gracefully and returns an appropriate error status. | | |
| **Expected Result** | API gracefully handles SQL Server connectivity issues and returns an appropriate error status. | | |
| **Actual Result** |  | | |
| **Pass/Fail** |  | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 26/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Reference** | 12 | **Project Name** | Travel Diary |
| **Test Case Id** | 12.2 | **Test Type** | Backend |
| **Test Case Description** | Test for SQL injection prevention. | | |
| **Test Steps** | 1. Attempt to inject SQL statements through API parameters. 2. Verify that the API correctly prevents SQL injection attacks. | | |
| **Expected Result** | API prevents SQL injection attempts and responds appropriately. | | |
| **Actual Result** |  | | |
| **Pass/Fail** |  | | |
| **Date Prepared** | 22/1/24 | | |
| **Date Run** | 26/1/24 | | |
| **Prepared By** | Bushra Ahmed | | |
| **Tested By** | Bushra Ahmed | | |