Test Procedure #1: User Registration

1. **Purpose**: This procedure verifies the functionality of the user registration system.
2. **Test Environment Setup**: Ensure the application is running and accessible. Prepare a set of valid and invalid user credentials for testing.
3. **Test Cases**:
   - *TC-001*: Verify that a user can sign up with valid credentials.
   - *TC-002*: Verify that a user cannot sign up with invalid credentials.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Reset the state of any user accounts that were locked during testing.

Test Procedure #2: User Authentication

1. **Purpose**: This procedure verifies the functionality of the user authentication system.
2. **Test Environment Setup**: Ensure the application is running and accessible. Prepare a set of valid and invalid user credentials for testing.
3. **Test Cases**:
   - *TC-003*: Verify that a user can sign in with valid credentials.
   - *TC-004*: Verify that a user cannot sign in with invalid credentials.
   - *TC-005*: Verify that an admin can sign in with valid credentials.
   - *TC-006*: Verify that an admin cannot sign in with invalid credentials.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Log out any users and admin that were logged in during the tests.

Test Procedure #3: Password Reset Functionality

1. **Purpose**: This procedure verifies the functionality of the password reset system.
2. **Test Environment Setup**: Ensure the application is running and accessible. Prepare a set of valid user accounts for testing.
3. **Test Cases**:
   - *TC-001*: Verify that a user can sign up with valid credentials.
   - *TC-007*: Verify that a user can reset their password with valid credentials.
   - *TC-008*: Verify that a user cannot reset their password with invalid credentials.
   - *TC-003*: Verify that a user can sign in with the new password after the reset.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Reset the user's password to the original password after the tests.

Test Procedure #4: Workspace Create Functionality

1. **Purpose**: This procedure verifies the functionality of the workspace creation system by an admin.
2. **Test Environment Setup**: Ensure the workspace booking application is running and accessible. Prepare a set of valid admin accounts for testing.
3. **Test Cases**:
   - *TC-003*: Verify that an admin can sign in to the admin panel.
   - *TC-051*: Verify that an admin can access the workspace creation form in the admin panel.
   - *TC-009*: Verify that an admin can fill out the workspace creation form with valid data and submit the workspace creation form.
   - *TC-010*: Verify that an admin cannot fill out the workspace creation form with invalid data and submit the workspace creation form.
   - *TC-054*: Verify that the new workspace is correctly added to the system and is visible to users for booking.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Remove any workspaces created during the tests to keep the test environment clean.

Test Procedure #5: Workspace Update Functionality

1. **Purpose**: This procedure verifies the functionality of the workspace update system by an admin.
2. **Test Environment Setup**: Ensure the workspace booking application is running and accessible. Prepare a set of valid admin accounts and existing workspaces for testing.
3. **Test Cases**:
   - *TC-003*: Verify that an admin can sign in to the admin panel.
   - *TC-051*: Verify that an admin can access the list of existing workspaces in the admin panel.
   - *TC-011*: Verify that an admin can update the details of the workspace using the update form with valid data.
   - *TC-012*: Verify that an admin cannot update the details of the workspace using the update form with invalid data.
   - *TC-054*: Verify that the updated workspace details are correctly reflected in the system and are visible to users for booking.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Revert any workspace updates made during the tests to their original state.

Test Procedure #6: Workspace Delete Functionality

1. **Purpose**: This procedure verifies the functionality of the workspace deletion system by an admin.
2. **Test Environment Setup**: Ensure the workspace booking application is running and accessible. Prepare a set of valid admin accounts and existing workspaces for testing.
3. **Test Cases**:
   - *TC-003*: Verify that an admin can sign in to the admin panel.
   - *TC-051*: Verify that an admin can access the list of existing workspaces in the admin panel.
   - *TC-013*: Verify that an admin can select a workspace to delete from the list and confirm the deletion of the selected workspace.
   - *TC-054*: Verify that the deleted workspace is no longer available in the system and is not visible to users for booking.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Ensure that any workspaces deleted during the tests are restored if necessary for future tests.

Test Procedure #7: Workspace Reservation Create Functionality

1. **Purpose**: This procedure verifies the functionality of the workspace reservation system.
2. **Test Environment Setup**: Ensure the workspace booking application is running and accessible. Prepare a set of valid workspaces and user accounts for testing.
3. **Test Cases**:
   - *TC-003*: Verify that a user can sign in into the system.
   - *TC-050*: Verify that a user can view available workspaces.
   - *TC-062*: Verify that a user can select a workspace for reservation.
   - *TC-014*: Verify that a user can set a valid reservation date and time for a selected workspace and confirm a workspace reservation.
   - *TC-015*: Verify that a user cannot set a invalid reservation date and time for a selected workspace and confirm a workspace reservation.
   - *TC-023*: Verify that a user can view their reservations in the profile.
   - *TC-026*: Verify that a workspace shows as unavailable in the calendar once it has been reserved.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Cancel any reservations made during the tests and ensure the workspaces are available for future tests.

Test Procedure #8: Workspace Reservation Update Functionality

1. **Purpose**: This procedure verifies the functionality of the workspace reservation update system.
2. **Test Environment Setup**: Ensure the workspace booking application is running and accessible. Prepare a set of valid workspaces, user accounts, and existing reservations for testing.
3. **Test Cases**:
    - *TC-003*: Verify that a user can sign in into the system.
    - *TC-023*: Verify that a user can view their existing workspace reservations in the profile.
    - *TC-018*: Verify that a user can update the reservation setting the valid date and time.
    - *TC-019*: Verify that a user cannot update the reservation setting the invalid date and time.
    - *TC-023*: Verify that the updated reservation details are correctly reflected in the system.
    - *TC-026*: Verify that a workspace shows as unavailable in the calendar once it has been reserved at a new date.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Revert any reservation updates made during the tests to their original state.

Test Procedure #9: Workspace Reservation Payment Functionality

1. **Purpose**: This procedure verifies the functionality of the payment system for workspace reservations.
2. **Test Environment Setup**: Ensure the workspace booking application is running and accessible. Prepare a set of valid user accounts, existing workspaces, and reservations for testing.
3. **Test Cases**:
    - *TC-003*: Verify that a user can sign in into the system.
    - *TC-023*: Verify that a user can view their existing workspace reservations in the profile.
    - *TC-021*: Verify that a user can select a reservation to pay for from the list and get an invoice to the email.
    - *TC-023*: Verify that the payment is correctly processed and the reservation is marked as paid in the system.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Refund any payments made during the tests and ensure the reservations are available for future tests.

Test Procedure #10: Workspace Reservation Delete Functionality

1. **Purpose**: This procedure verifies the functionality of the reservation deletion system.
2. **Test Environment Setup**: Ensure the workspace booking application is running and accessible. Prepare a set of valid user accounts and existing workspace reservations for testing.
3. **Test Cases**:
   - *TC-003*: Verify that a user can sign in into the system.
   - *TC-023*: Verify that a user can view their existing workspace reservations in the profile.
   - *TC-020*: Verify that a user can select a reservation to delete from the list and confirm the deletion.
   - *TC-023*: Verify that the deleted reservation is no longer listed in the user's reservations.
   - *TC-026*: Verify that the deleted reservation is no longer available in the system and is not visible to other users.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Ensure that any reservations deleted during the tests are restored if necessary for future tests.

Test Procedure #11: Workspace Reservation Cancel Functionality

1. **Purpose**: This procedure verifies the functionality of the reservation cancellation system.
2. **Test Environment Setup**: Ensure the workspace booking application is running and accessible. Prepare a set of valid user accounts and existing workspace reservations for testing.
3. **Test Cases**:
   - *TC-003*: Verify that a user can sign in into the system.
   - *TC-023*: Verify that a user can view their existing workspace reservations in the profile.
   - *TC-022*: Verify that a user can select a reservation to cancel from the list and confirm the cancellation of the selected reservation.
   - *TC-023*: Verify that the canceled reservation is no longer listed in the user's reservations.
   - *TC-026*: Verify that the workspace from the canceled reservation is available for booking by other users.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Ensure that any reservations canceled during the tests are restored if necessary for future tests.

Test Procedure #12: User Profile Update Functionality

1. **Purpose**: This procedure verifies the functionality of the user profile update system.
2. **Test Environment Setup**: Ensure the application is running and accessible. Prepare a set of valid user accounts for testing.
3. **Test Cases**:
   - *TC-003*: Verify that a user can sign in into the system.
   - *TC-082*: Verify that a user can edit their personal information (such as name and surname) on the profile page with valid data.
   - *TC-083*: Verify that a user cannot edit their personal information (such as name and surname) on the profile page with invalid data.
   - *TC-084*: Verify that the updated personal information is correctly reflected in the system and visible to the user upon refreshing the page or logging out and back in.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Revert any profile updates made during the tests to their original state.

Test Procedure #13: Admin Profile Update Functionality

1. **Purpose**: This procedure verifies the functionality of the admin profile update system.
2. **Test Environment Setup**: Ensure the application is running and accessible. Prepare a set of valid admin accounts for testing.
3. **Test Cases**:
   - *TC-003*: Verify that an admin can sign in into the system.
   - *TC-085*: Verify that an admin can edit their personal information (such as name and surname) on the profile page with valid data.
   - *TC-086*: Verify that an admin cannot edit their personal information (such as name and surname) on the profile page with invalid data.
   - *TC-087*: Verify that the updated personal information is correctly reflected in the system and visible to the admin upon refreshing the page or logging out and back in.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Revert any profile updates made during the tests to their original state.

Test Procedure #14: User Password Change Functionality

1. **Purpose**: This procedure verifies the functionality of the password change system in a user profile.
2. **Test Environment Setup**: Ensure the application is running and accessible. Prepare a set of valid user accounts for testing.
3. **Test Cases**:
   - *TC-003*: Verify that a user can sign in into the system.
   - *TC-088*: Verify that a user can access the change password form on the profile page and fill out the change password form with a new valid password.
   - *TC-089*: Verify that a user can access the change password form on the profile page and cannot fill out the change password form with a new invalid password.
   - *TC-003*: Verify that the user's password is updated in the system and the user can log in with the new password.
   - *TC-004*: Verify that the old password is no longer valid after the change.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Reset the user's password to the original password after the tests.

Test Procedure #15: Admin Password Change Functionality

1. **Purpose**: This procedure verifies the functionality of the password change system in an admin profile.
2. **Test Environment Setup**: Ensure the application is running and accessible. Prepare a set of valid admin accounts for testing.
3. **Test Cases**:
   - *TC-003*: Verify that an admin can sign in into the system.
   - *TC-088*: Verify that an admin can access the change password form on the profile page and fill out the change password form with a new valid password.
   - *TC-089*: Verify that an admin can access the change password form on the profile page and cannot fill out the change password form with a new invalid password.
   - *TC-003*: Verify that the admin's password is updated in the system and the user can log in with the new password.
   - *TC-004*: Verify that the old password is no longer valid after the change.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Reset the admin's password to the original password after the tests.

Test Procedure #16: User Profile Delete Functionality

1. **Purpose**: This procedure verifies the functionality of the user profile deletion system.
2. **Test Environment Setup**: Ensure the application is running and accessible. Prepare a set of valid user accounts for testing.
3. **Test Cases**:
   - *TC-003*: Verify that a user can sign in into the system.
   - *TC-094*: Verify that a user can access their user profile page, request to delete their profile and confirm the deletion.
   - *TC-095*: Verify that the user's profile is no longer available in the system and the user can no longer log in with the deleted profile's credentials.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Ensure that any profiles deleted during the tests are restored if necessary for future tests.

Test Procedure #17: User Profile Logout Functionality

1. **Purpose**: This procedure verifies the functionality of the user profile logout system.
2. **Test Environment Setup**: Ensure the application is running and accessible. Prepare a set of valid user accounts for testing.
3. **Test Cases**:
   - *TC-003*: Verify that a user can sign in to their account.
   - *TC-096*: Verify that a user can logout and is redirected to the main page.
   - *TC-003*: Verify that the user cannot access their profile without logging in again.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Ensure that any changes made during the tests are reverted to their original state.

Test Procedure #18: Admin Profile Logout Functionality

1. **Purpose**: This procedure verifies the functionality of the user profile logout system.
2. **Test Environment Setup**: Ensure the application is running and accessible. Prepare a set of valid user accounts for testing.
3. **Test Cases**:
   - *TC-003*: Verify that an admin can sign in to their account.
   - *TC-097*: Verify that an admin can logout and is redirected to the main page.
   - *TC-003*: Verify that the admin cannot access their profile without logging in again.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Ensure that any changes made during the tests are reverted to their original state.

Test Procedure #19: Workspace Search Functionality

1. **Purpose**: This procedure verifies the functionality of the workspace search system.
2. **Test Environment Setup**: Ensure the workspace booking application is running and accessible. Prepare a set of valid user accounts and existing workspaces for testing.
3. **Test Cases**:
   - *TC-040*: Verify that a guest can access the workspace search function and get results match the entered valid search criteria.
   - *TC-041*: Verify that a guest can access the workspace search function and cannot get results according to the entered invalid search criteria.
   - *TC-003*: Verify that a user can sign in into the system.
   - *TC-043*: Verify that a user can access the workspace search function and get results match the entered valid search criteria.
   - *TC-044*: Verify that a user can access the workspace search function and cannot get results according to the entered invalid search criteria.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Ensure that any changes made during the tests are reverted to their original state.

Test Procedure #20: Workspace Reservation Date Selection Functionality

1. **Purpose**: This procedure verifies the functionality of the date selection system for workspace reservations.
2. **Test Environment Setup**: Ensure the workspace booking application is running and accessible. Prepare a set of valid user accounts and existing workspaces for testing.
3. **Test Cases**:
   - *TC-025*: Verify that a user can access the workspace reservation function.
   - *TC-028*: Verify that a user can access the date selection function for the selected workspace.
   - *TC-029*: Verify that a user can select a forward date from the date selection function.
   - *TC-030*: Verify that a user can select a backward date from the date selection function
   - *TC-031*: Verify that a user can select a current date from the date selection function.
   - *TC-025*: Verify that the selected date with its reservations are correctly reflected in the reservation form.
4. **Order of Execution**: Execute the test cases in the order listed.
5. **Cleanup**: Ensure that any changes made during the tests are reverted to their original state.