# Complete Test Case Documentation

This document contains all test cases collaboratively designed and executed for the Edges Course Center software.

## TestCase 1

Description: Testing the Add Account Functionality with valid input

Pre-requisites: New user form filled correctly

Test Inputs:  
Name: Ahmed  
Age: 26  
DOB: 23/10/1998  
Education: Masters\_Student  
Gender: Male  
Username: EdgesAcademy  
Password: Edges123  
Password Recheck: Edges123

Expected Output: The DB should be updated with the previous inputs correctly

Reason: All inputs are in correct format

## TestCase 2

Description: Testing the Add Account Functionality with mismatched passwords

Pre-requisites: New user form with incorrect password recheck

Test Inputs:  
Name: Rania  
Age: 54  
DOB: 31/12/1970  
Education: PHD\_Holder  
Gender: Female  
Username: EdgesAcademy2024  
Password: Edges\_123\_  
Password Recheck: Edges\_123\_$$

Expected Output: The DB should not be updated due to password mismatch

Reason: Passwords do not match

## TestCase 3

Description: Test Admin Login Success

Pre-requisites: Admin credentials are correct

Test Inputs:  
Admin Username: Mohamed Tarek, Token: 10203040

Expected Output: Admin login should be successful

Reason: Valid token provided

## TestCase 4

Description: Test Admin Login Failure due to incorrect token

Pre-requisites: Admin token is invalid

Test Inputs:  
Admin Username: Mohamed Tarek, Token: Wrong values x3

Expected Output: Admin login should fail after 3 attempts

Reason: Security check failed

## TestCase 5

Description: Test Normal User Login Success

Pre-requisites: Credentials are correct

Test Inputs:  
Username: EdgesAcademy, Password: Edges123

Expected Output: User should login successfully

Reason: Correct credentials

## TestCase 6

Description: Test Login Failure due to wrong password

Pre-requisites: Incorrect password input

Test Inputs:  
Username: EdgesAcademy, Password: WrongPassword

Expected Output: User should not be logged in

Reason: Incorrect password

## TestCase 7

Description: Test Login Failure due to wrong username

Pre-requisites: Username does not exist

Test Inputs:  
Username: NonExistentUser, Password: SomePassword

Expected Output: Login should fail due to invalid username

Reason: Username is invalid

## TestCase 13

Description: Reserve a valid course successfully

Pre-requisites: AdminUser2 is logged in and course is available

Test Inputs:  
Course ID: 1

Expected Output: User should be enrolled in course ID 1

Reason: Valid credentials and valid course ID

## TestCase 14

Description: Prevent enrolling in the same course twice

Pre-requisites: User is already enrolled

Test Inputs:  
Course ID: 1 again

Expected Output: Enrollment should fail with AlreadyEnrolled

Reason: Duplicate enrollment

## TestCase 15

Description: Prevent enrolling in out-of-range course ID

Pre-requisites: Course ID is invalid

Test Inputs:  
Course ID: 99

Expected Output: Enrollment should fail

Reason: Invalid course ID

## TestCase 16

Description: Show enrolled courses for user

Pre-requisites: User has enrolled courses

Test Inputs:  
ShowStudentCourses()

Expected Output: User should see their enrolled course(s)

Reason: Basic functionality test

## TestCase 17

Description: User deletion removes course reservations

Pre-requisites: User is deleted after course enrollment

Test Inputs:  
Delete user after enrollment

Expected Output: Enrollments for user should be cleared

Reason: Data consistency test

## TestCase 19

Description: Admin adds a new normal user

Pre-requisites: Admin mode active, correct form

Test Inputs:  
Name: Omar  
Age: 30  
DOB: 15/8/1994  
Education: Graduate  
Gender: Male  
Username: TestUser\_Omar  
Password: OmarPass123

Expected Output: User is added to DB

Reason: Simulates admin using Add user feature

## TestCase 20

Description: Admin deletes an existing user

Pre-requisites: Valid user created

Test Inputs:  
User ID of newly created user

Expected Output: User should be removed from DB

Reason: Admin deletion functionality

## TestCase 21

Description: Admin attempts to delete non-existent user

Pre-requisites: Use out-of-range ID

Test Inputs:  
User ID = 100

Expected Output: Delete\_Account should return FALSE

Reason: Boundary test

## TestCase 22

Description: Admin prints all users

Pre-requisites: DB has users

Test Inputs:  
Call DBM\_PrintUsers()

Expected Output: Should print all users without crashing

Reason: Output verification

## TestCase 23

Description: Admin prints one valid user

Pre-requisites: User ID 0 exists

Test Inputs:  
user\_id = 0

Expected Output: Print and return TRUE

Reason: Valid query

## TestCase 24

Description: Admin prints invalid user ID

Pre-requisites: User ID = 150 (out of range)

Test Inputs:  
Call DBM\_PrintUserData\_Admin(150)

Expected Output: Should return FALSE

Reason: Error handling