##### University Department

##### Information System

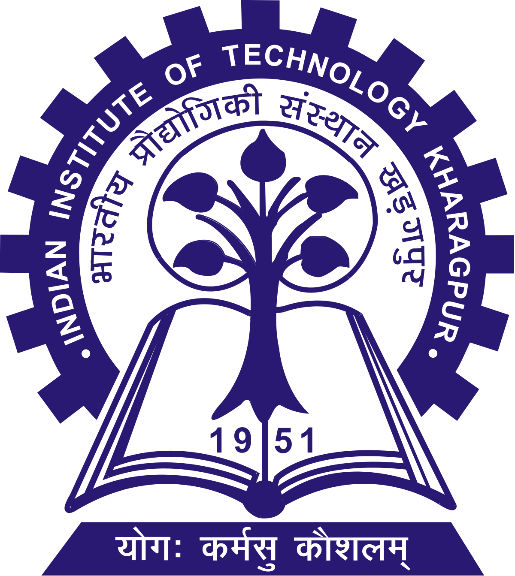
Testing Plan

*Prepared by*

*Anirudh M(21CS10004)*

*Aviral Pandey(21CS10089)*

*Mayukha Marla(21CS10041)*



**INDIAN INSITUTUTE OF TECHNOLOGY KHARAGPUR**

**Contents**

We have the following scenarios for testing.

* 1. Signing in to the system (as different kind of users)
  2. Adding a new student to the system
  3. Searching for a student, using both roll number and name queries
  4. Viewing student performance and grade details
  5. Registering courses for a particular student
  6. Entering grades for a student
  7. Adding new items to the department inventory
  8. Searching inventory items using type/name
  9. Viewing and adding new courses
  10. Viewing and adding new publications
  11. Viewing and adding new projects and changing their status
  12. Adding funds to the Department Account inventory and withdrawing funds from the Department Account

**Description**

In the following section, we provide the description for the unit tests of various files.

1. **Login Page (Filename login.html)**

In this, we test whether the login functionality works correctly. This is tested on the following credentials.

* 1. Login ID correct and Password incorrect (Error)
  2. Login ID incorrect and Password correct (Error)
  3. Login ID incorrect and Password incorrect (Error)
  4. Any entry left blank, either Login ID or Password (Error)
  5. Login ID correct and Password correct (Correct)

Error cases show respective error messages. Correct case successfully logs in the user.

In case user has forgotten password, we test the forgot password functionality as follows. The user must click on forgot password, then submit their email id. Then they will receive an email with a link. The link will redirect them to reset password page where they can reset new password.

1. **Home Page(Filename home.html)**

Check login option. Check sign out button and try to login again. The user must have to login again and cannot sign in without entering credentials.

1. **Student Main Page (Filename StudentMain.html)**

Check all the buttons and options, and whether they lead to their respective webpages or not.

1. **View Student Page (Filename StudentView.html)**

Student should be able to check their profile which must include:

* + 1. Correct name, correct roll number, correct address, correct year of joining and the CGPA (these details can be manually checked) etc.
    2. Performance Section

1. Backlogged subjects should be shown in a different section.
2. SGPA of each semester should be computed accurately up to two decimal places. Already completed courses should also be visible.
3. Subjects which have been backlogged once but have been cleared should display an asterisk (\*).
4. **Register Courses Page (Filename StudentCourses.html)**

We need testing for the following:

* 1. Testing the Binding Action
     1. Cleared courses must not be shown.
     2. Backlog courses must be shown. This is because backlog courses can be registered again.
     3. Cleared Backlog courses must not be shown.
     4. A maximum of 12 courses can be registered in each semester, if a student has never had a backlog
     5. A maximum of eight courses can be registered in each semester if a student has ever had a backlog
     6. The courses, which have been just registered and not graded, must have R grade against them in the database.
  2. Searching for courses by name or course code.

1. **Secretary Main Screen (Filename SecretaryMain.html)**

Check all the buttons, and whether they lead to their respective screens or not.

1. **Entering Student Grades Page (Filename StudentGrades.html)**

Secretary should be able to add grades for each student. We need testing for the following:

* 1. **Testing the Binding Action**
     1. Only the ongoing courses are to be fetched.
     2. The subjects can be graded only after filling all the grades, otherwise, alerts must be shown.
     3. The grades for the cleared subjects should be updated and visible in the performance section of student view page.
     4. Correct cgpa should be calculated after entering all the grades. It should also be updated in the performance section of students view page.

1. **Adding New Student (Filename StudentNew.html)**

In this page, the secretary creates and adds a new student. Following things need to be checked:

* 1. Roll number format checking
     1. Correct format entered (<Last 2 year digits><2 department character identifiers><course identifier digit/character><4 serial number digits> )
     2. Any other format entered (error), alerts must be shown to enter the correct format roll number.
  2. Year of admission checking
     1. Year entered in the correct format, i.e. a 4 digit number
     2. Year entered in an incorrect format, i.e. some alphabets or special characters inserted or the year entered is not a 4 digit number (error)
  3. Duplicate Roll number checking (error)
  4. Some fields are left blank (error), respective fields shown in alerts.
  5. Course identifier matches with the course, otherwise alert shown.
  6. All entries have been entered correctly and the add button clicked, and then check if the said student can be accessed from the database (test case passed).

1. **Department Inventory (Filename Inventory.html)**

This is the inventory section of the department where all the items and their locations are mentioned.

* 1. **Searching for an item**
     1. **Blank name type all**

All items in the inventory are shown, this can be verified by counting the entries in the database.

* + 1. **Name specific type all**

Only the items whose names match with the entered name across all types should be shown.

* + 1. **Name blank type specific**

Only the items that match the specific type should be shown.

* + 1. **Name specific type specific**

Items that match with a particular name **and** the type selected should be shown.

* 1. **Adding an item to the inventory**
     1. **Transaction Failure (Error)**

1. Insufficient funds in the Department Account, i.e. this transaction will lead to negative balance.
2. Quantity of items is not positive or not a number
3. Price is not positive or not a number
4. One of the fields is blank(like location, type etc)
   * 1. **Transaction Succeeds**
5. Price and Quantity positive
6. All the blanks filled.
7. If location, name and type match with a particular item from the inventory exactly, increase quantity.
8. If either one of location, name and type do not match, add a new entry into the inventory.
9. **Department Account Page (Filename Account.html)**

**Adding money to the account**

* 1. **Money Successfully Inserted into the account**

1. All transaction fields filled using valid data.
2. Amount is positive and a number.
3. Date format correct (DD/MM/YYYY in this case).
4. Inserted money should show in the account passbook with the date, donor details and the current balance.
   1. **Money not inserted (Error)**
5. Donation amount is not positive or not a number (alert shown)
6. Any one field left blank
7. Date format incorrect (alert displaying the correct format of the date shown to the user)
8. Invalid Date (like 30 Feb 2021)

**Viewing cash book account details**

The output of this query should include the income, the expenditure and the balance.

1. **Department Courses (Filename DepartmentCourses.html)**

In this section, we check the functionality of adding new courses to the department.

* 1. Searching for Courses
     1. Blank Input – All courses of the department must be shown
     2. Course name entered– All courses of the department having names which are similar to the entered course name must be shown.
     3. Course code entered- The course having this unique course code must be shown.
  2. Adding new Courses
     1. **Course Successfully added**
        1. Course code not already present.
        2. Credits for the course should be a number and must not exceed a reasonable value (2-5) in this case.
        3. Newly added course available for registration and visible in when searching.
     2. **Course not Added (error)**
        1. Course with the same code already present.
        2. Credits entered is not a number
        3. Credits entered is a negative or a large number exceeding five.
        4. Either one of name, code or credits column left blank.

1. **Department Publication Section Screen (Filename DepartmentPublications.html)**

In this section, we test the various publications in the department. There is only one scenario, **adding a new publication**.

* + 1. **Publication Successfully added** 
       1. No field left blank
       2. The set of fields entered do not match with any previous data entry.
       3. Publication visible in the Publication Section screen.
    2. **Publication not added (error)**
       1. One or more of the fields left blank.
       2. The set of fields already match with a previous data entry.

1. **Department Projects (Filename DepartmentProjects.html)**
   1. **Adding a new Project**
      1. **Project Successfully Added** 
         1. All the fields must be filled (professor name, funds needed etc)
         2. Funds field is a valid integer and the requisite funds are already present in the bank account
         3. No project with the same name exists
      2. **Project Not Added (Exception Path)**
         1. One or more of the fields vacant.
         2. Funds is not an integer
         3. Funds is negative
         4. Sufficient funds not present in the bank account
         5. Project name already exists before
   2. **Testing the Binding Action to change status of the project**
      * 1. Status can be changed if the current status is ‘Ongoing’ to ‘Completed’
        2. Status cannot be changed if the current status is ‘Completed’
2. **Querying Student info page(Filename StudentSearch.html)**
   1. **Searching for a student**

The different scenarios include:

* 1. **Empty roll no. empty name**

In the case of nothing filled, all the students in the database must be shown. This can be verified by checking the number of students in total.

* 1. **Empty roll no. filled name**
  2. **Filled roll no. and filled name**

In each of the 2 cases above, all the students, which have names and roll numbers similar to the search queries (i.e. substrings), must be displayed

* 1. **Filled roll no. and filled name**

In this case, the screen will display all the students whose roll numbers match the roll number mentioned **or** whose names match the names mentioned.

* 1. **Testing the Binding Action**

Clicking on the data entry of each student must open a new popup, whose title must match the roll number of the student. All details of the student such as name, roll number, address, department, year of joining etc must be shown on this popup.