Type of test areas

- User interface
    - Look and feel
    - Responsive
    - Components
    - Navigation (Links)
    - Pages layout
    - Browser compatibility
    - Page titles
    - Active page
    - User experience
    - Content and spelling
    - Scrolling behavior
    - Consistency
- Functionality
    - Session management
    - Database saving changes and retrieving
    - User verification (login and registration)
    - Functional requirements are fulfilled
    - Data verification
    - Correct images
    - Form Validation
    - Confirmation, error, and warning messages
    - Calculations
    - File upload verification
    - Back button

Test Cases Checklist

Layout

- □ Pages that represent the same entity look the same or similar across the applications.
- □ Each header has the same style and content.
- □ Each footer has the same style and content
- □ The page title reflects the form/page content.
- □ The page title contains no redundant or unnecessary words like Maintain or Information.
- □ The page title is logically consistent with the menu path required to reach the page.

Usability

- □ The webpages flow should be easy to understand.
- □ Navigation bar should be provided in each page and is consistent and similar in all pages.
- □ Home link should be there on every single page.

Content

- □ Content should be readable and there should be no grammatical or spelling errors
- □ Informational, Warning and Error messages should be professional, clear and into the point.
- □ Icons used are representative.
- □ Fonts should be the same throughout the webpages according to the specifications

□ Check favicon is present on the tab bar

□ used colors should be meaningful

for example:

- success messages should be displayed in green.
- Error messages should be displayed in red.
- Warning messages in yellow.

Database saving changes and retrieving

- Inserting a record:

□ You can enter values only in fields that should be accessible.

□ You cannot Save after leaving one or more functionally required fields blank. (Try one field at a time to make sure that the form forces you to enter each required field.)

□ You can successfully Save after entering values into each functionally required field and leaving all optional fields blank.

□ You can Save after entering a value into every field.

□ You can fill each field to the maximum field length and save the record. No field is longer than its corresponding database column.

□ The form checks for duplicate primary keys or unique index values before inserting records into the database. Verify that you can change a duplicate key and save successfully.

□ All column values are correct e.g., valid email format, phone numbers, national Id … etc.

- Selecting a record:

□ The form retrieves the correct records.

□ When you query a record, the form retrieves a value for each field if the relevant database column is available.

□ When you query each part of the page, it displays records in a logical order.

- Updating a record:

□ You can update all fields that should allow update.

□ You can save the fields correctly in the database on update.

□ You can erase the values for each optional field in a record then Save the changes. Query that records to verify your changes saved.

□ You can update the values for each required field, making sure that it's not left empty and display a message when a required field is left empty.

□ When the application design allows users to update the primary key or unique index values of a record, the form checks for duplicate values.

- Deleting a record:

□ If a block allows deletion, a confirmation message always appears when you try to delete a record. If you choose not to delete the record, the delete does not occur.

□ When you delete a record, it disappears from the screen.

□ When you delete a record from the form and save, the record disappears from the database.

Form Validation

- □ Fields have the correct length.
- □ All enterable text items have a label or a placeholder.
- □ Display of the fields not excessively crowded, uses white spaces.
- □ Verify that labels are in the correct position.
- □ The form has a balanced layout.
- □ No text strings hard coded into the form/page, including error messages.
- □ Fields alignment is consistent.
- □ An asterisk sign should be displayed for all mandatory fields.

- Clear Form
  - □ When you have modified one or more fields of the form, Clear Form asks you to save your changes before clearing the form.
  - □ Clear button returns every block to its original state, where every field contains either blanks or a default value in the case of insert and the original data from the database in the case of update.
  - □ When you have not made any changes to the relevant block(s), Clear button does not ask you to save or abandon your changes before clearing the form.

- Select field
  - □ Select options should show only valid values.
  - □ Select field offers the correct choices in a logical order.
  - □ Select options shows fields at a reasonable size without truncated values.
  - □ Select fields always have a default value. They may include a blank value and blank may be the default if blank is a valid value, however.
  - □ all the options data is arranged in chronological order.

- Numeric Field Validation
  - □ You cannot enter alphabetic characters such as “A”.
  - □ You can enter a numeric value that falls between a field's lower and upper limits.
  - □ You cannot enter a numeric value less than a field's lower limit or greater than its upper limit.
  - □ You can type numbers of the appropriate precision into a numeric field.
  - □ You cannot enter a number that exceeds the precision of the underlying database column.

- Date Fields
  - □ You can type only dates into a date field.
  - □ The Calendar is available on all date fields.
  - □ The Calendar displays only the Date fields if the user cannot specify a time with the date.
  - □ The Calendar displays the Time fields if the user can specify a time with the date.
  - □ Date and Time format must be consistent across the webpages.
  - □ Start and End Date, end date must be greater than start date.
  - □ Start and End time, end time must be greater than start time.
  - □ Data validation, birthdate should be validated not to be in the future.
  - □ Test that leap years are validated correctly & do not cause errors.

- Check Boxes

  □ Check boxes have a reasonable default value (on or off, as will most likely be the case).

  □ The user can check multiple checkboxes (Each checkbox has different name attribute)

- Radio buttons

  □ Radio buttons have a reasonable default value (on or off, as will most likely be the case).

  □ Verify use of radio group where one of the radio buttons can be selected.

- Percentage Fields

  □ You cannot enter percentages over 100 or less than 0.

- Currency Fields

  □ You can enter currency values with an adequate number of digits before and after the decimal point.

  □ Changing the currency code changes the formatting of the currency amount to be appropriate to the new currency.

- Buttons

  □ In general, buttons are navigable. Exceptions are buttons enabled only while in a particular field and clear buttons

  □ The look and feel of the button changes when the user hovers over the button.

  □ The button should be easily found. The most important button should be identified and visualized more clearly compared to the less important ones.

  □ Each button initiates the correct action or brings up the correct page.

- Response Time

  □ When you query the form, it retrieves a record in five seconds or less (or if that is not possible, displays the "watch" cursor or a progress bar).

  □ Field-to-field navigation requires less than one second.

  □ You can save the changes in five seconds or less (or if that is not possible, displays the "watch" cursor or a progress bar).