Feature 1: Creating a project

A user cannot create a project without including these fields:

1. Name of project
2. Team members
3. Project deadline (to create progress bar)

Specific test cases:

1. Test that project can not be created without one of the mandatory fields. Enter any combination of 1 or 2 fields and attempt to create a project. Attempt should fail
    a. Do not allow the submission of letters in the project deadline field.
    b. Allow submission of date without day (mm/yy or dd/mm/yy) format.
    c. A message will inform the user of which fields are missing/insufficient
2. Test that project is successfully created and shows up on the home page when all three fields are entered.

These tests will be tested locally through a sql query:

INSERT INTO projects (name, members, deadline)

VALUES (‘presentation’, ‘group 1’, ‘12/1/25’)

Then we will ensure that this project is accurately reflected in Ragan and Ishita’s homepage.

User acceptance: Have a user create an account, create a team, then create a project and add their team to it. They should be able to see the project and interact with its components

Feature 2: Creating a team

In order to create a team the user must include:

1. A unique team name
2. Team lead
3. At least two users

Specific test cases:

1. Test that a team is created and stored in our system when a user puts in a unique team name, a team lead, and at least two users. When you go to the teams page the new team should show up and the users on the team will have a tag signifying that they are members of the team.
2. Test that an attempt to create a team fails if:
    a. the team name is already used within the project or there is no team name
    b. if there is no team lead or the team lead is not a member of the project
    c. if there are less than two users being added to the team or any of the members are not a member of the project

       A message will inform the user of which fields are missing/insufficient

Information from the team creation will be stored in the teams database. We will also have a table that bridges team ids and project member ids. We will update this database with the new team id and member id connections.

These tests will be tested locally through sql query:

INSERT INTO teams (projectName, teamLead) values (“testTeam”, “Ragan”)
After the query is run we can check if the changes are reflected on the teams page, and in the user data of the team members.
User acceptance: A user will create a team, insert a team name, team lead, and team members and upon submitting the team should be able to view their team on the teams page and see tags on the team members tasks page signifying what teams they are on.

Feature 3: Interacting with the whiteboard
A user cannot interact with the whiteboard associated with a project if they are not logged in or are not part of the project
Specific test cases:

1. Test that the user can add “sticky notes” of text to the whiteboard.
2. Test that the user can delete “sticky notes” of text on the whiteboard
3. Test that the user can comment on “sticky notes” of text on the whiteboard
4. Test that user can resolve comments
5. Test that user can move items around the whiteboard
6. Test that items cannot be moved off of the whiteboard
7. Test that the tool bar interacts appropriately with the whiteboard
    a. Create sticky note creates text box
    b. Delete deletes a sticky note
    c. Move moves notes around
    d. Comment creates a comment box

These tests will be tested on the website itself. One of the developers will test each feature as they add the functionality.
We will ensure that all of these features function adequately and as intended.
User acceptance: Have a user create an account and join a project. Then have them interact with the whiteboard. They should be able to use the full functionality of the whiteboard with no errors

Feature 4: Calendar
In order for a calendar to populate the home page of a users account, they must have a valid and active account (not recently deleted, account has all required fields in database: name, username, email, password)
In order to insert a new event into the calendar the user must include:

1. Date
2. Event Title
3. Name of Project event is for

4. Comments field (optional)

Specific test cases:

1. Test that a calendar event is created and stored in our system when a user puts in a data, event name, an associated project, and an optional comments field. When you go to your calendar you should be able to see the event.
2. Test that an attempt to create a calendar event fails if:
    a. The date is not specified or is an invalid date (in the past)
    b. An event name is not specified
    c. A project is not selected
    A message will inform the user of which fields are missing/insufficient

Information from the event creation will be stored in the calendar database.

These tests will be tested locally through sql query:

```
INSERT INTO calendar (date, event, project, comments) values (“1/1/25”, “New Year”,
“SampleProject”, “New year new me”)
```

After the query is run we can check if the changes are reflected on the calendar. We can also ensure that these changes were accepted locally by running:

```
SELECT event FROM calendar WHERE date = “1/1/25”
```

User acceptance: A user will create a calendar event, specifying the date, event name, project, and comments. The new event should be reflected in the calendar.