

2305028

Agile approach to software development

__/_/___

output: leave are approved or disapproved
output 2: leave approval/disapproval notification is sent to the employee.

Test case for HR # 3

Precondition: 1) Has to be employee

2) Has to be logged in

3) application record entrance is pending

Procedure step 1 → HR view emp id

step 2 → HR enters the details to the Database

output: the leave record is entered to the database.

सनायका के विनोद 24.

Test case for employee #1

Precondition: ① should be an employee
② should be logged in

Procedure: step 1 → employee enters emp-id
step 2 → enters emp name
step 3 → selects the type of leave
step 4 → selects the start and end date
step 5 → adds additional note.

output 1 → if leave quota is limit then application goes to manager

output 2 → if leave quota exceeds then warning pops up then also it still goes to the manager

Test case for manager #2

Precondition: ① should be an employee
② should be logged in
③ Application is Pending

Procedure: Step 1 → Manager views emp ID, type of leave, start date/end date.

Step 2: verify searched limit,

Step 3: Manager clicks on emp ID to further clarify the reason

Step 4: part 1: manager approves the leave
part 2: manager rejects the leave.

Part

Company

1) Application for leave (reason, dates) forwarded to manager, HR, Tech lead (Tech lead can be optional)
Whoever approves, track goes to HR (make a diff module to keep a track)

2) Recruitment module: find the vacancies of company website (description, roles). To apply make module to get the info.

3) Salary Disbursement

4) Approval: employee can send their work to be reviewed. manager can make employee too.

5) Registration / login



HR View:

Emp id:

Emp Name:

January ☒

Days Present:

Days Absent:

Leave Taken type ☒

Leave Quota: x / x

		Agile	Trad. waterfall
		Team is cross functional	Roles are static
		Client Involvement	With client involvement
		Better work flow management	Concepted work flow
		less documentation	more documentation
		Fast testing phase	Slow and extensive testing phase
		Quicker feedback	no feedback
		Revisit previous phase	No revisiting phase
		<p>i) Roles for were interchanged while working in agile allowing exploring market potential.</p> <p>ii) with ability to revisit phase & regular client involvement is phase feasible.</p> <p>iii) Focusing on one module at a time with its designing and testing before moving to other module</p> <p>iv) Testing while module is properly designed provided better test cases</p> <p>v) With client involvement feedbacks were constant and beneficial</p> <p>vi) Flexible work nature of agile allowed revisit to previous phase if need be</p>	

```

$ git checkout master
Switched to branch 'master'

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git merge inhuman2
Auto-merging 28.txt
CONFLICT (content): Merge conflict in 28.txt
Automatic merge failed; fix conflicts and then commit the result.

student@BCA37 MINGW64 ~/Desktop/28 (master|MERGING)
$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecmerge p4merge araxis bc codecompare smerge emerge vimdiff nvimdiff
Merging:
28.txt

Normal merge conflict for '28.txt':
  {local}: modified file
  {remote}: modified file
Hit return to start merge resolution tool (vimdiff):
4 files to edit

student@BCA37 MINGW64 ~/Desktop/28 (master|MERGING)
$ git commit -m "final merge"
[master 7f2d69c] final merge

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ cat 28.txt
hello me
bye me

meet me 2day
meet me 2morrow

```

```

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ cat 28.txt
hello me
bye me

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git checkout Human
error: pathspec 'Human' did not match any file(s) known to git

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git branch Human

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git checkout Human
Switched to branch 'Human'

student@BCA37 MINGW64 ~/Desktop/28 (Human)
$ ls
28.txt

student@BCA37 MINGW64 ~/Desktop/28 (Human)
$ vi 28.txt

student@BCA37 MINGW64 ~/Desktop/28 (Human)
$ cat 28.txt
hello me
bye me
meet me

student@BCA37 MINGW64 ~/Desktop/28 (Human)
$ git checkout master
Switched to branch 'master'
M      28.txt

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git checkout Human
Switched to branch 'Human'
M      28.txt

student@BCA37 MINGW64 ~/Desktop/28 (Human)

```

```
MINGW64:/c/Users/student/Desktop/28
Switched to branch 'Human'
M      28.txt

student@BCA37 MINGW64 ~/Desktop/28 (Human)
$ git status
On branch Human
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   28.txt

no changes added to commit (use "git add" and/or "git commit -a")

student@BCA37 MINGW64 ~/Desktop/28 (Human)
$ git commit -m "human branch"
On branch Human
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   28.txt

no changes added to commit (use "git add" and/or "git commit -a")

student@BCA37 MINGW64 ~/Desktop/28 (Human)
$ git add 28.txt

student@BCA37 MINGW64 ~/Desktop/28 (Human)
$ git commit -m "human branch"
[Human e018bbd] human branch
1 file changed, 1 insertion(+)

student@BCA37 MINGW64 ~/Desktop/28 (Human)
$ cat 28.txt
hello me
bye me
meet me

student@BCA37 MINGW64 ~/Desktop/28 (Human)
$ git checkout master
Switched to branch 'master'
```



```
student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git merge human
Updating f6d7bf6..e018bbd
Fast-forward
 28.txt | 1 +
 1 file changed, 1 insertion(+)

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ cat 28.txt
hello me
bye me
meet me

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git branch inhuman

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git branch inhuman2

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git checkout inhuman
Switched to branch 'inhuman'

student@BCA37 MINGW64 ~/Desktop/28 (inhuman)
$ vi 28.txt

student@BCA37 MINGW64 ~/Desktop/28 (inhuman)
$ git add 28.txt

student@BCA37 MINGW64 ~/Desktop/28 (inhuman)
$ git commit -m "inhuman branch"
[inhuman 012646b] inhuman branch
 1 file changed, 1 insertion(+), 1 deletion(-)

student@BCA37 MINGW64 ~/Desktop/28 (inhuman)
$ git checkout master
Switched to branch 'master'
```

```
student@BCA37 MINGW64 ~/Desktop/28 (Higher)
$ cat 28.txt
hello me
bye me

student@BCA37 MINGW64 ~/Desktop/28 (Higher)
$ git add 28.txt
warning: LF will be replaced by CRLF in 28.txt.
The file will have its original line endings in your working directory

student@BCA37 MINGW64 ~/Desktop/28 (Higher)
$ git commit
[Higher f6d7bf6] "branch commit"
1 file changed, 1 insertion(+)

student@BCA37 MINGW64 ~/Desktop/28 (Higher)
$ cat 28.txt
hello me
bye me

student@BCA37 MINGW64 ~/Desktop/28 (Higher)
$ git checkout master
Switched to branch 'master'

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ cat 28.txt
hello me

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git merge Higher
Updating 7244620..f6d7bf6
Fast-forward
 28.txt | 1 +
1 file changed, 1 insertion(+)

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ cat 28.txt
hello me
bye me
```

MINGW64:/c/Users/student/Desktop/28

```
student@BCA37 MINGW64 ~/Desktop/28 (inhuman)
git checkout master
Switched to branch 'master'
```

```
student@BCA37 MINGW64 ~/Desktop/28 (master)
git merge inhuman
Updating e018bbd..012646b
Fast-forward
 28.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

```
student@BCA37 MINGW64 ~/Desktop/28 (master)
git checkout inhuman2
Switched to branch 'inhuman2'
```

```
student@BCA37 MINGW64 ~/Desktop/28 (inhuman2)
vi 28.txt
```

```
student@BCA37 MINGW64 ~/Desktop/28 (inhuman2)
git add
nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .'
hint: Turn this message off by running
hint: "git config advice.addEmptyPathsSpec false"
```

```
student@BCA37 MINGW64 ~/Desktop/28 (inhuman2)
git add 28.txt
```

```
student@BCA37 MINGW64 ~/Desktop/28 (inhuman2)
git commit -m "inhuman2 branch"
inhuman2 269f2bd] inhuman2 branch
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
student@BCA37 MINGW64 ~/Desktop/28 (inhuman2)
git checkout master
Switched to branch 'master'
```

```
student@BCA37 MINGW64 ~/Desktop/28 (master)
git merge inhuman2
Auto-merging 28.txt
```



```
student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git config --global user.email "your@example.com"

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git config --global user.email "HigherHuman"

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git commit -m "1st commit"
[master (root-commit) 7244620] 1st commit
1 file changed, 1 insertion(+)
create mode 100644 28.txt

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git log
commit 72446207174305b7cf696867e00da76b33c7b2ca (HEAD -> master)
Author: student <HigherHuman>
Date:   Wed Apr 9 12:29:43 2025 +0530

    1st commit

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git branch
* master

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git branch Higher

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git checkout Higher
Switched to branch 'Higher'

student@BCA37 MINGW64 ~/Desktop/28 (Higher)
$ vi 28.txt

student@BCA37 MINGW64 ~/Desktop/28 (Higher)
$ cat 28.txt
hello me
```

```

student@BCA37 MINGW64 ~/Desktop
$ cd 28

student@BCA37 MINGW64 ~/Desktop/28
$ git innit
git: 'innit' is not a git command. See 'git --help'.

The most similar command is
    init

student@BCA37 MINGW64 ~/Desktop/28
$ git init
Initialized empty Git repository in C:/Users/student/Desktop/28/.git/

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ vi 28.txt

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    28.txt

nothing added to commit but untracked files present (use "git add" to track)

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git add 28.txt
warning: LF will be replaced by CRLF in 28.txt.
The file will have its original line endings in your working directory

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git status
On branch master

No commits yet

```

```

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   28.txt

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git commoit -m "1st commit"
git: 'commoit' is not a git command. See 'git --help'.

The most similar command is
    commit

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git commit -m "1st commit"
Author identity unknown

*** Please tell me who you are.

Run

    git config --global user.email "you@example.com"
    git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'student@BCA37.(none)')

student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git config --global user.email "your@example.com"

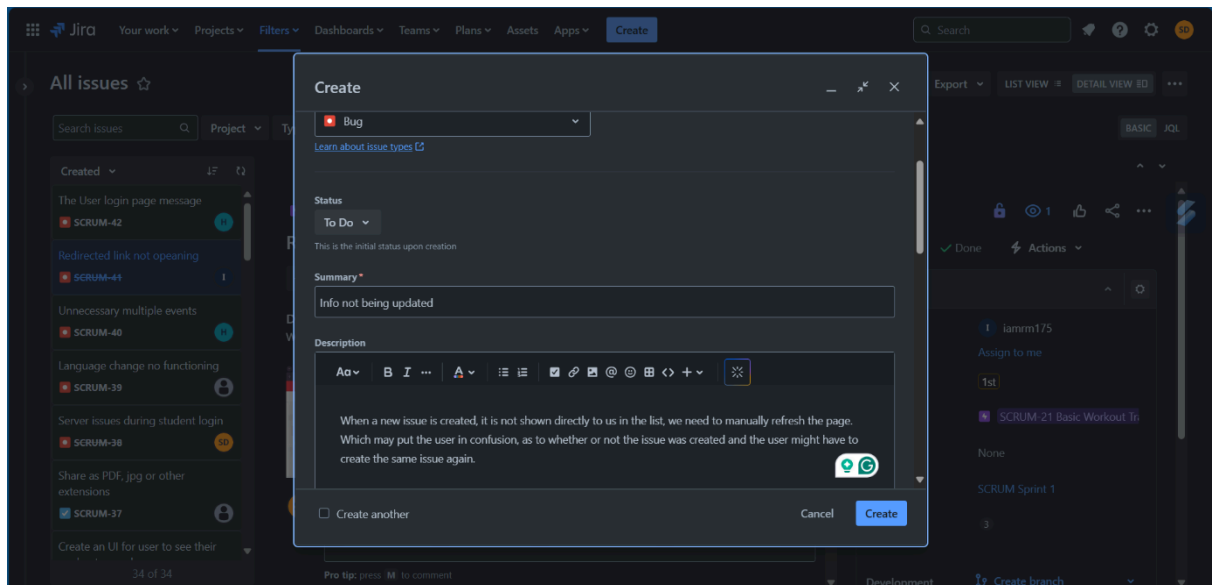
student@BCA37 MINGW64 ~/Desktop/28 (master)
$ git config --global user.email "HigherHuman"

student@BCA37 MINGW64 ~/Desktop/28 (master)

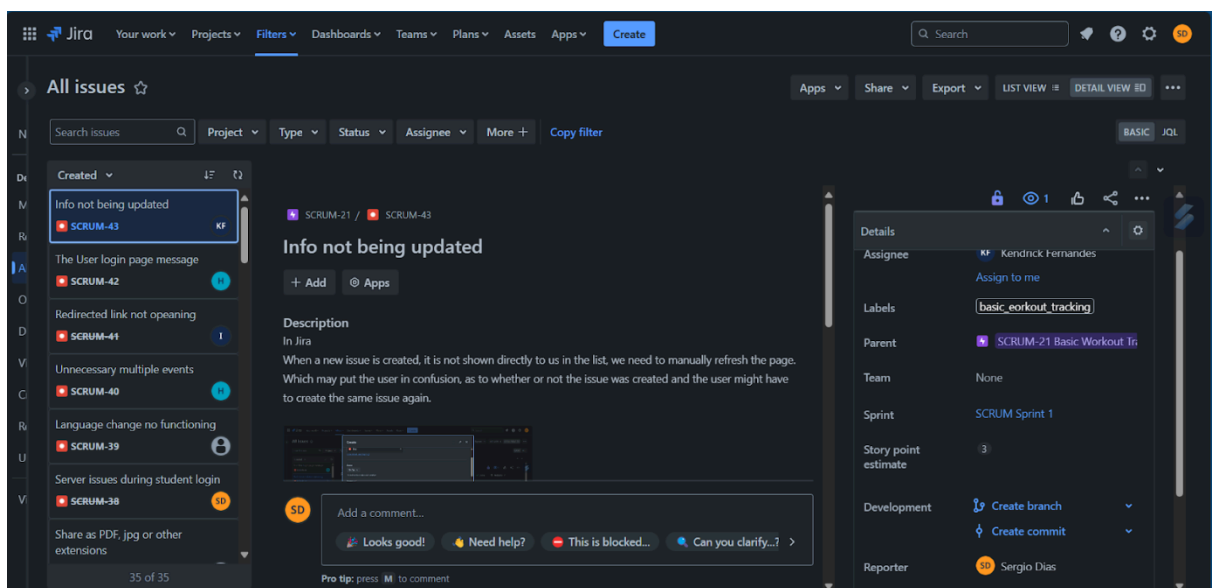
```

Jira- Fitness App

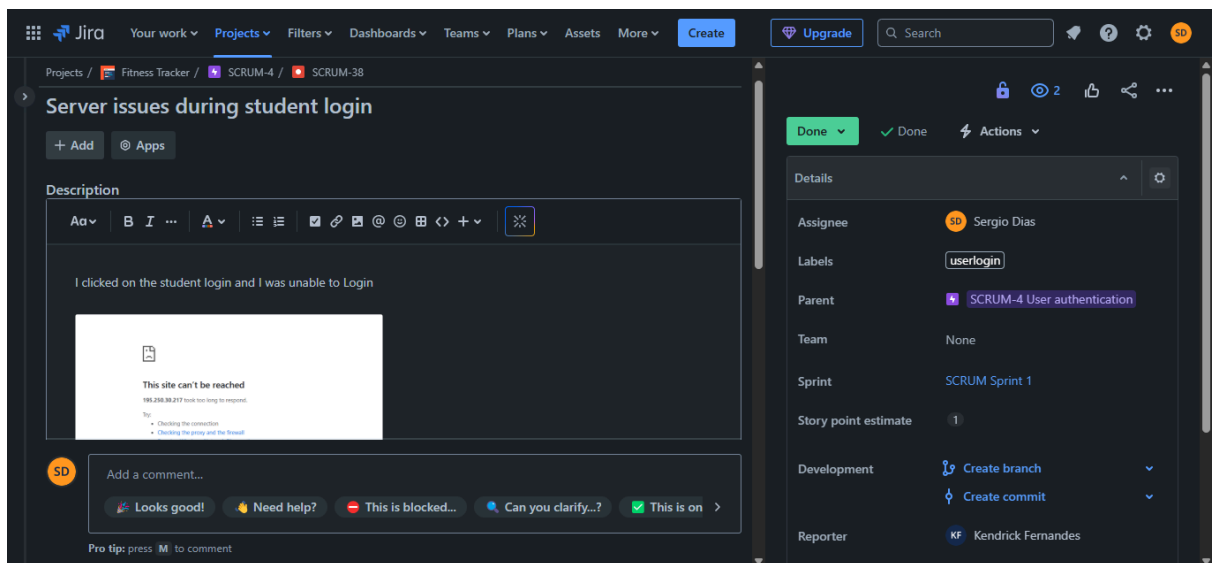
1. Creating issues



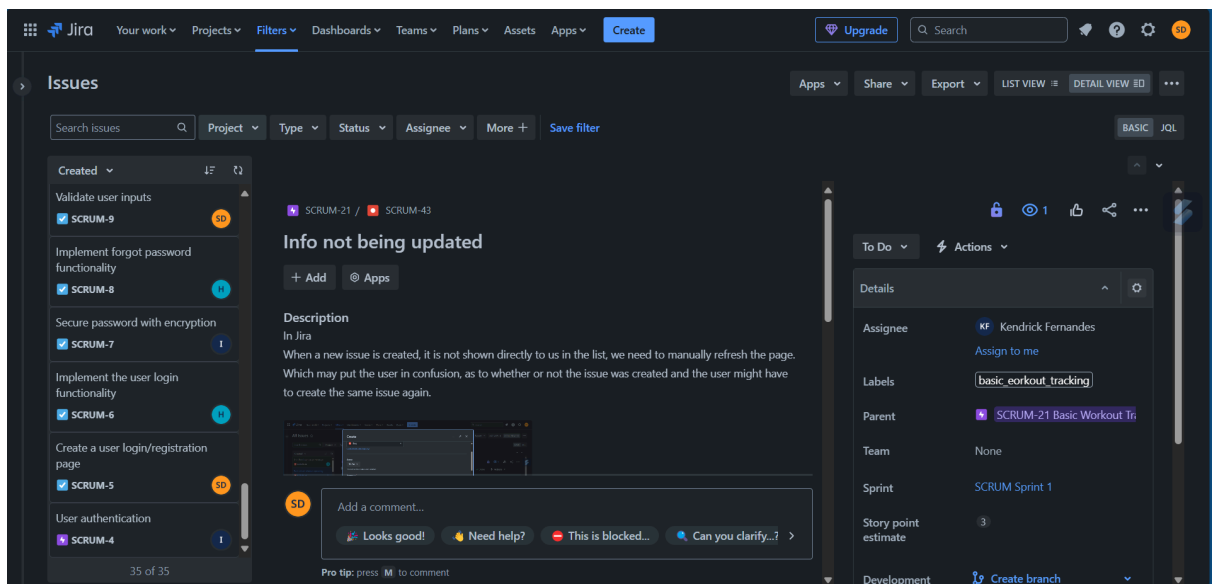
2. Assigned task



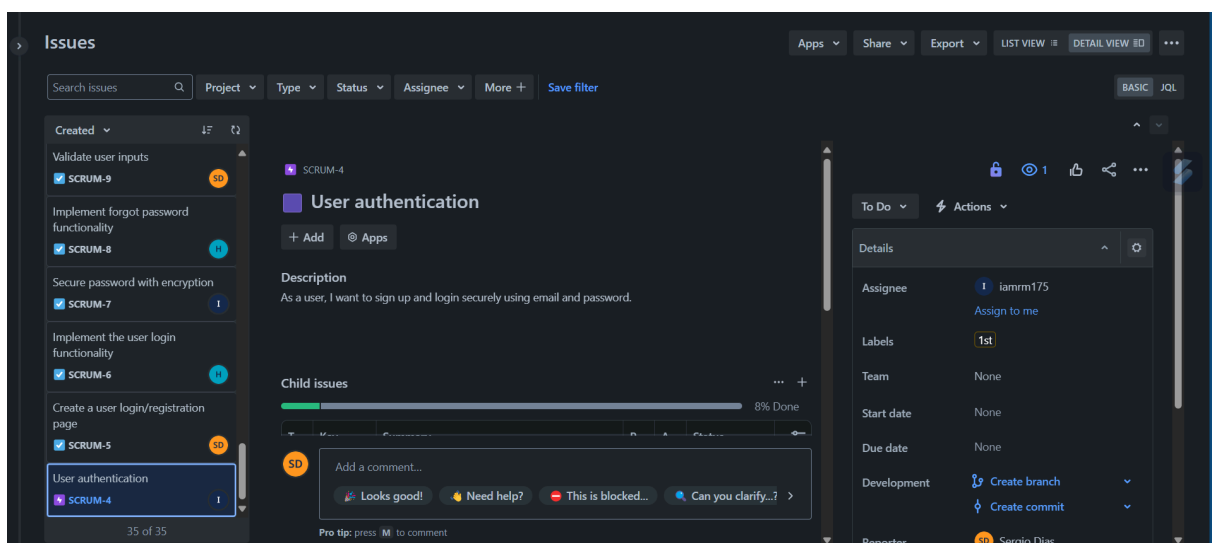
3. Status changed!



4. List of issues



5. Assigned and work in 'to do' status



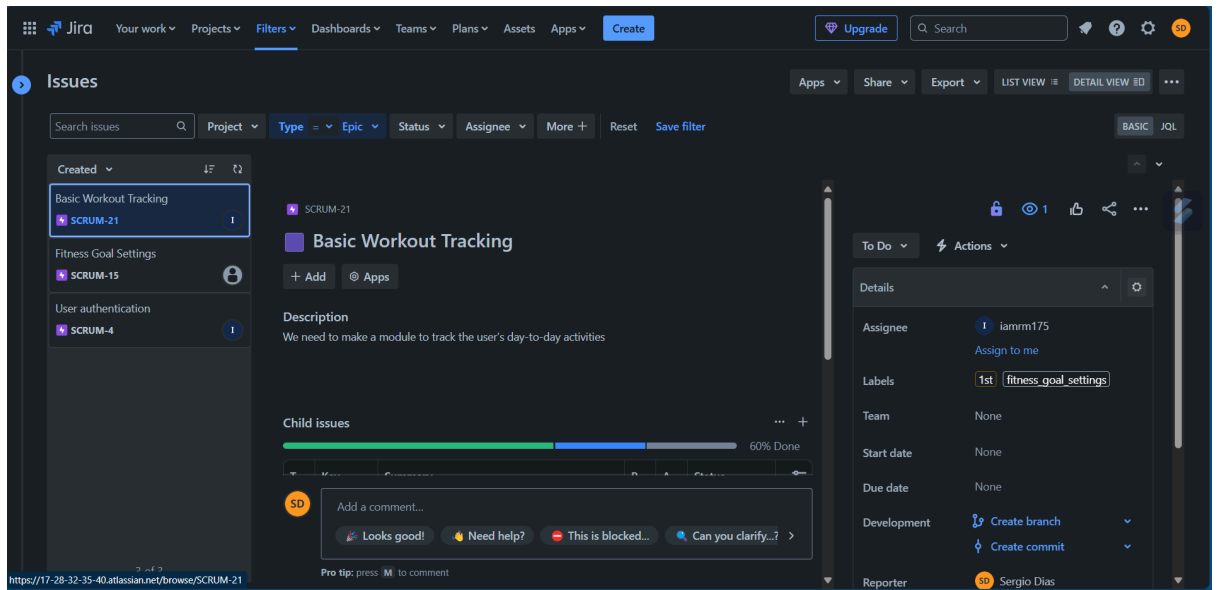
6. Work completed by the member

The screenshot shows a Jira issue page for SCRUM-21. The issue is titled "logo it asks to redirect and opens it in new tab but the site fails to load". The description includes a video player showing a user interface with a logo and a redirect button. The video player shows a user clicking a button, which leads to a redirect page. The description also mentions that the site fails to load after the redirect. The issue is assigned to iamrm175 and has a label "1st". The parent issue is SCRUM-21 Basic Workout Tracking. The team is None, and the sprint is SCRUM Sprint 1. The story point estimate is 3. The development status is "Create branch" and "Create commit". The reporter is Sergio Dias.

7. Completed 2 epics

The screenshot shows a Jira Issues page. The left sidebar lists issues: Basic Workout Tracking (SCRUM-21), Fitness Goal Settings (SCRUM-15), User authentication (SCRUM-4), and SCRUM-4. The main content area shows the details of SCRUM-15. The description is "In this feature the user can Plan his goals, modify his goals as per his needs, and wants". The child issues table shows two issues: SCRUM-22 (As a user I want to set goals for my activities to increase my fitness) and SCRUM-23 (Design an UI to add current body details). Both child issues are marked as "DONE". The issue is assigned to Unassigned and has a label "1st fitness_goal_settings". The team is None, and the start and due dates are None. The development status is "Create branch" and "Create commit". The reporter is Sergio Dias.

8. 3rd epic in progress



Product/Output:-

fitness (body setting)

Task 1

Add Body Details

* Enter your Age :-

* Enter your Weight :-

* Enter your Height :-

Next →

Task 2

Select Your Tasks :-

Task 1	<input type="radio"/>
Task 2	<input type="radio"/>
Task 3	<input type="radio"/>
Task 4	<input type="radio"/>
Task 5	<input type="radio"/>
Task 6	<input type="radio"/>
Task 7	<input type="radio"/>
Task 8	<input type="radio"/>

(NEXT) (BACK)

Task 3

SET YOUR GOALS

Task 1 :- set goals

Task 2 :- set goals

Task 3 :- set goals

Confirm

User2

Task1

Edit Your Tasks :-

Task 1	<input type="radio"/>
Task 2	<input type="radio"/>
Task 3	<input type="radio"/>
Task 4	<input type="radio"/>
Task 5	<input type="radio"/>
Task 6	<input type="radio"/>
Task 7	<input type="radio"/>
Task 8	<input type="radio"/>

(Next) (Back)

Edit Your Goals

Task 1 :- edit goals

Task 2 :- edit goals

Task 3 :- edit goals

Summary Page

Summary

Total reps done

1. Pushups: 120 / 1000

2. Squats: 200 / 1000

3. Dips: 50 / 100

4. Pulls: 50 / 100

5. Plank: 10 / 100

6. Burpees: 10 / 100

7. Mountain Climbers: 10 / 100

8. Side Plank: 10 / 100

9. Russian Twists: 10 / 100

10. High Kicks: 10 / 100

11. Jumping Jacks: 10 / 100

12. Lunges: 10 / 100

13. Sit-ups: 10 / 100

14. Crunches: 10 / 100

15. Leg Raises: 10 / 100

16. Deadlifts: 10 / 100

17. Bench Press: 10 / 100

18. Shoulder Press: 10 / 100

19. Bicep Curls: 10 / 100

20. Tricep Dips: 10 / 100

21. Core Exercises: 10 / 100

22. Cardio: 10 / 100

23. Stretching: 10 / 100

24. Meditation: 10 / 100

25. Sleep: 10 / 100

26. Hydration: 10 / 100

27. Nutrition: 10 / 100

28. Consistency: 10 / 100

29. Motivation: 10 / 100

30. Persistence: 10 / 100

31. Focus: 10 / 100

32. Discipline: 10 / 100

33. Determination: 10 / 100

34. Resilience: 10 / 100

35. Endurance: 10 / 100

36. Strength: 10 / 100

37. Speed: 10 / 100

38. Agility: 10 / 100

39. Balance: 10 / 100

40. Coordination: 10 / 100

41. Reflexes: 10 / 100

42. Sensory: 10 / 100

43. Motor: 10 / 100

44. Cognitive: 10 / 100

45. Emotional: 10 / 100

46. Social: 10 / 100

47. Cultural: 10 / 100

48. Environmental: 10 / 100

49. Technological: 10 / 100

50. Scientific: 10 / 100

51. Historical: 10 / 100

52. Geographical: 10 / 100

53. Political: 10 / 100

54. Economic: 10 / 100

55. Legal: 10 / 100

56. Medical: 10 / 100

57. Educational: 10 / 100

58. Religious: 10 / 100

59. Philosophical: 10 / 100

60. Artistic: 10 / 100

61. Musical: 10 / 100

62. Literary: 10 / 100

63. Cinematic: 10 / 100

64. Televisual: 10 / 100

65. Digital: 10 / 100

66. Virtual: 10 / 100

67. Augmented: 10 / 100

68. Mixed: 10 / 100

69. Immersive: 10 / 100

70. Interactive: 10 / 100

71. Collaborative: 10 / 100

72. Participatory: 10 / 100

73. Generative: 10 / 100

74. Adaptive: 10 / 100

75. Responsive: 10 / 100

76. Flexible: 10 / 100

77. Scalable: 10 / 100

78. Sustainable: 10 / 100

79. Resilient: 10 / 100

80. Robust: 10 / 100

81. Secure: 10 / 100

82. Reliable: 10 / 100

83. Efficient: 10 / 100

84. Effective: 10 / 100

85. Successful: 10 / 100

86. Profitable: 10 / 100

87. Valuable: 10 / 100

88. Meaningful: 10 / 100

89. Impactful: 10 / 100

90. Transformative: 10 / 100

91. Revolutionary: 10 / 100

92. Innovative: 10 / 100

93. Creative: 10 / 100

94. Imaginative: 10 / 100

95. Visionary: 10 / 100

96. Inspirational: 10 / 100

97. Motivational: 10 / 100

98. Empowering: 10 / 100

99. Enabling: 10 / 100

100. Facilitating: 10 / 100

START

Workout List

Brain Workout

1. Pushups	1x10
2. Squats	1x15
3. Dips	1x7
4. Pulls	1x10
5. Plank	2:00
6. Burpees	1x10
7. Squats	1x15

START →

During Workout

1 of 20

Push ups

1 x 10

(←) (→)

Post workout

* GREAT JOB!

* Workout Complete

Calories burned: 600

Total reps done:

1. Pushups	30
2. Squats	45
3. Dips	10
4. Pulls	10
5. Plank	10

Git Hub

```
student@danger MINGW64 ~/Desktop/practic
$ mkdir conn

student@danger MINGW64 ~/Desktop/practic
$ cd conn

student@danger MINGW64 ~/Desktop/practic/conn
$ git init
Initialized empty Git repository in C:/Users/student/Desktop/practic/conn/.git/

student@danger MINGW64 ~/Desktop/practic/conn (master)
$ git remote add H https://github.com/HigherHuman/high.git

student@danger MINGW64 ~/Desktop/practic/conn (master)
$ git pull H main
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (6/6), 1.75 KiB | 36.00 KiB/s, done.
From https://github.com/HigherHuman/high
* branch          main          -> FETCH_HEAD
* [new branch]     main          -> H/main

student@danger MINGW64 ~/Desktop/practic/conn (master)
```

Activate
Go to Setti

```
student@danger MINGW64 ~/Desktop/practic/conn (master)
$ ls
README.md  first.txt

student@danger MINGW64 ~/Desktop/practic/conn (master)
$ git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream H master

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

student@danger MINGW64 ~/Desktop/practic/conn (master)
$ vi first.txt

student@danger MINGW64 ~/Desktop/practic/conn (master)
$ cat first.txt
farst
secind
thurd
fawrth
fwifth
```

```

student@danger MINGW64 ~/Desktop/practic/conn (master)
$ git commit -m "first commit" first.txt
[master c5d5ec4] first commit
1 file changed, 2 insertions(+)

student@danger MINGW64 ~/Desktop/practic/conn (master)
$ git status
On branch master
nothing to commit, working tree clean

student@danger MINGW64 ~/Desktop/practic/conn (master)
$ git log
commit c5d5ec402bc80b8b7c31e6bf9726905acedf5e41 (HEAD -> master)
Author: jyoti kumari <nehadec0612@gmail.com>
Date: Wed Mar 19 11:36:00 2025 +0530

    first commit

commit 2a610ef2b0ecc26c3bf6d036651099a40ebdac30 (H/main)
Author: HigherHuman <manavkaushik2907@gmail.com>
Date: Wed Mar 19 11:09:45 2025 +0530

    Create first.txt

commit 5c97c1beef29e63ef3f49f25c84726aa112dfeb2
Author: HigherHuman <manavkaushik2907@gmail.com>

```

```

Initial commit

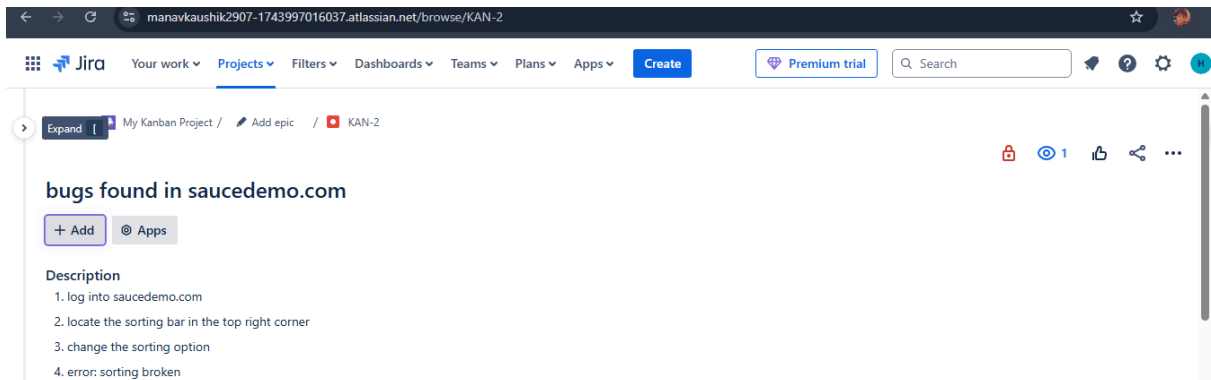
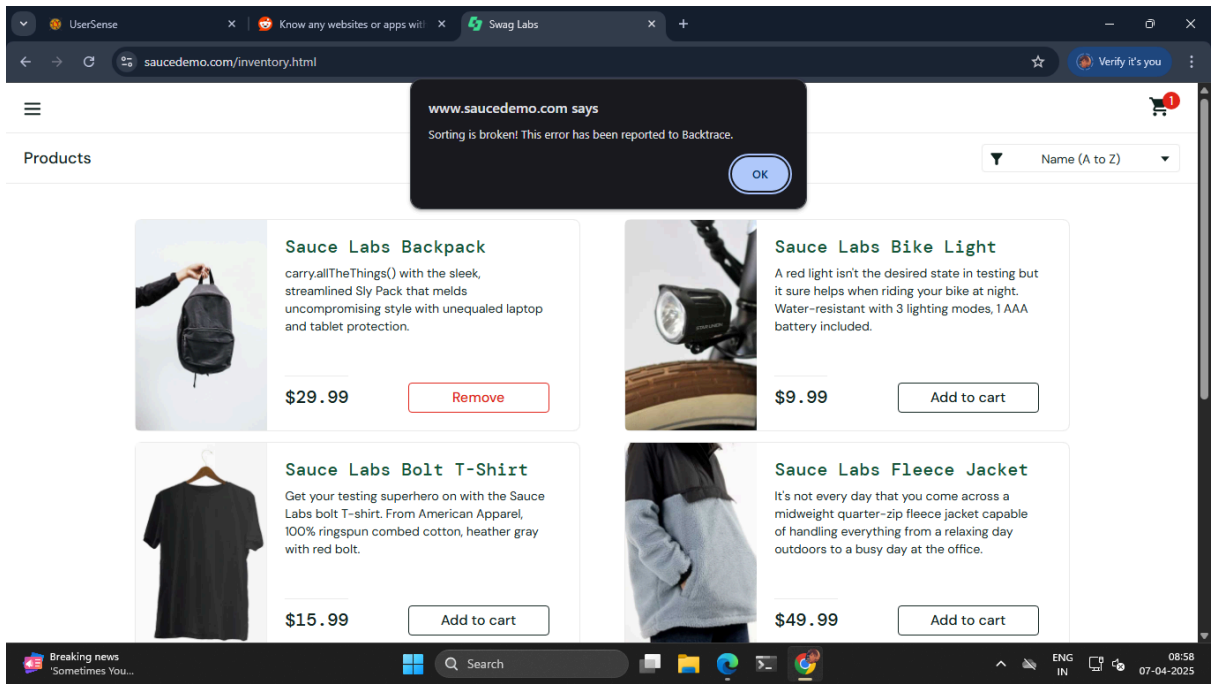
student@danger MINGW64 ~/Desktop/practic/conn (master)
$ git push H main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/HigherHuman/high.git'

student@danger MINGW64 ~/Desktop/practic/conn (master)
$ git push H master

git push H master

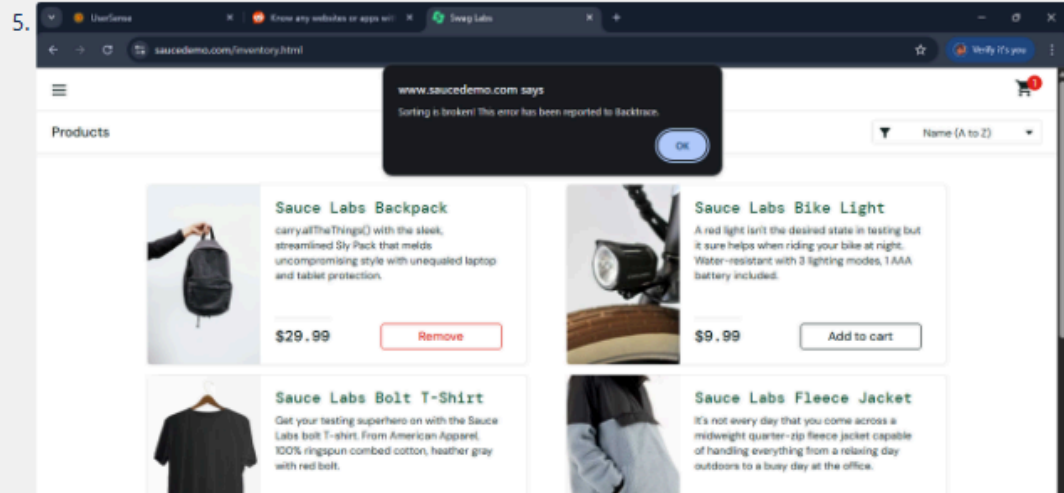
```

Creating issues for bugs in a website: saucedemo.com

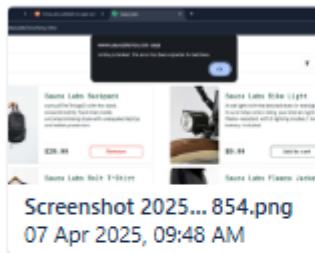


3. change the sorting option


4. error: sorting broken



Attachments 2



Confluence content ⓘ

 Known errors

JUnit test

```
package test;

public class Demo1
{
    int add(int a, int b)
    {
        return a+b;
    }
}
```

```

package test;

import static org.junit.jupiter.api.Assertions.*;

class Demo1Test {

    @Test
    void testAdd()
    {
        Demo1 d1=new Demo1();
        assertEquals(4,d1.add(2,2));

    }

}

```

Javadoc

The screenshot shows the Eclipse IDE interface with the Javadoc for the 'test' package generated. The left sidebar displays the project structure, with the 'test' package selected. The main editor area shows the Javadoc content, including navigation links, a search bar, and a list of classes.

Package test

package test

- Classes

Angular Language Server (org.eclipse.lsp4e.LanguageServerWrapper@469eeacc)

PACKAGE **CLASS** USE TREE INDEX SEARCH HELP

test > Demo1

Contents

Description

Constructor Summary

Method Summary

Constructor Details

Method Details

circle(double)

```
(package private) double circle(double r)
```

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Details

Demo1

```
public Demo1()
```

Method Details

circle

```
double circle(double r)
```

Parameters:

r - : excepts radius in double type

Selenium

```
1  /*
2   * right click on project
3   * properties
4   * java build path
5   * add reference libraries(selenium)
6   * add library
7   * jre system library
8   * alternate jre, installed , standard ym, directory
9   * tick off previous version of java
10  */
11
12  package sele;
13
14  import org.openqa.selenium.By;
15  import org.openqa.selenium.Keys;
16  import org.openqa.selenium.WebDriver;
17  import org.openqa.selenium.chrome.ChromeDriver;
18
19  public class Auto
20  {
21      public static void main(String[] args)
22      {
23          System.setProperty("webdriver.chrome.driver", "D:\\chromedriver-win64\\chromedriver.exe");//add chromedriver.exe in the end after cop
24          WebDriver driver=new ChromeDriver();
25          driver.navigate().to("https://github.com/login");
26          driver.findElement(By.name("login")).sendKeys("manavkaushik2907@gmail.com");
27          driver.findElement(By.name("password")).sendKeys("h");
28          driver.findElement(By.name("commit")).sendKeys(Keys.ENTER);
29      }
30  }
31
32  */
```

Chrome is being controlled by automated test software.



Sign in to GitHub

Incorrect username or password.



Username or email address

manavkaushik2907@gmail.com

Password

[Forgot password?](#)

Sign in

[Sign in with a passkey](#)

New to GitHub? [Create an account](#)