

Superlit User Manual For Teachers

Abstract

This document serves as a guide to the `superlit` platform. This guide is meant for teachers of a university.

Superlit is a code evaluation platform powered with smart AI features. It helps you conduct your coding assignments and labs drastically more efficiently. It makes a significant effort to avoid plagiarism and has features to uphold design principles in code.

Contact For Feature Requests & Bug Fixes

B S Anurag Rao

+91 9663006833

raoanu2004@gmail.com

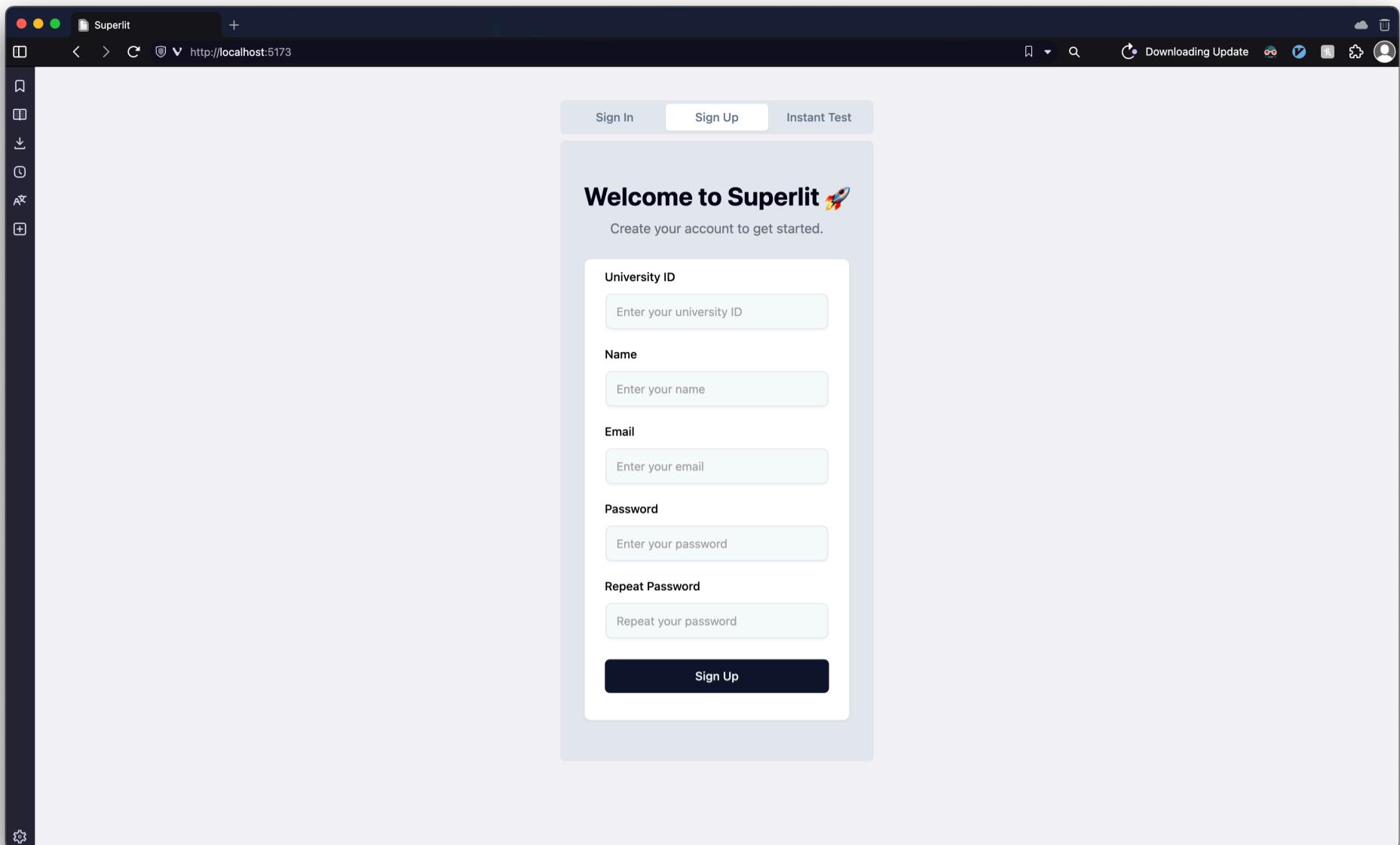
Terms Used & General Information

- The word `university ID` refers to the ID provided by the university. This can be employee ID for teachers and student ID for students.
-

Creating An Account

- Teachers must create an account through a hidden page on the website. You can open this hidden page by typing the following url into your browser:

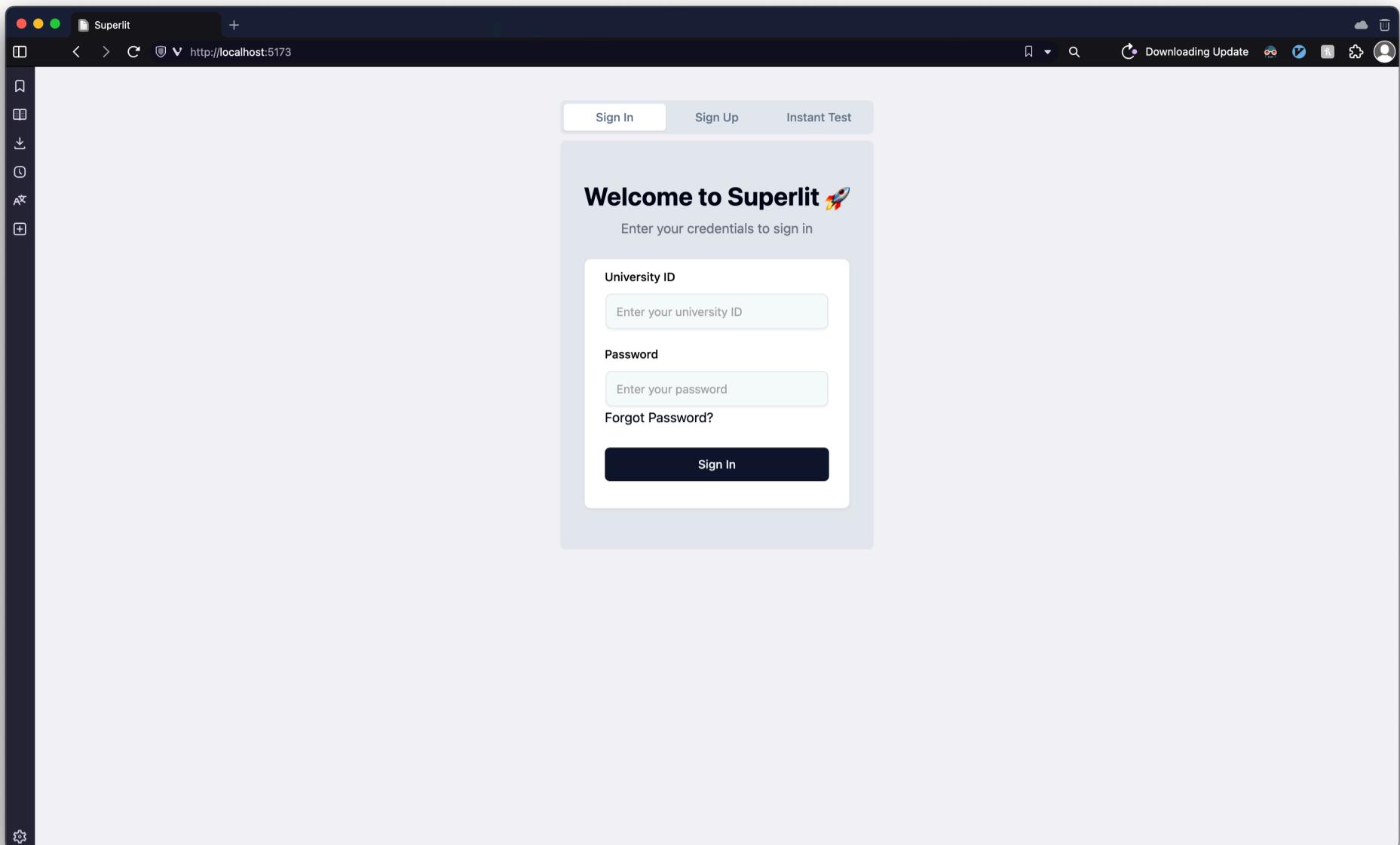
`http://10.2.80.90/signup/teacher`



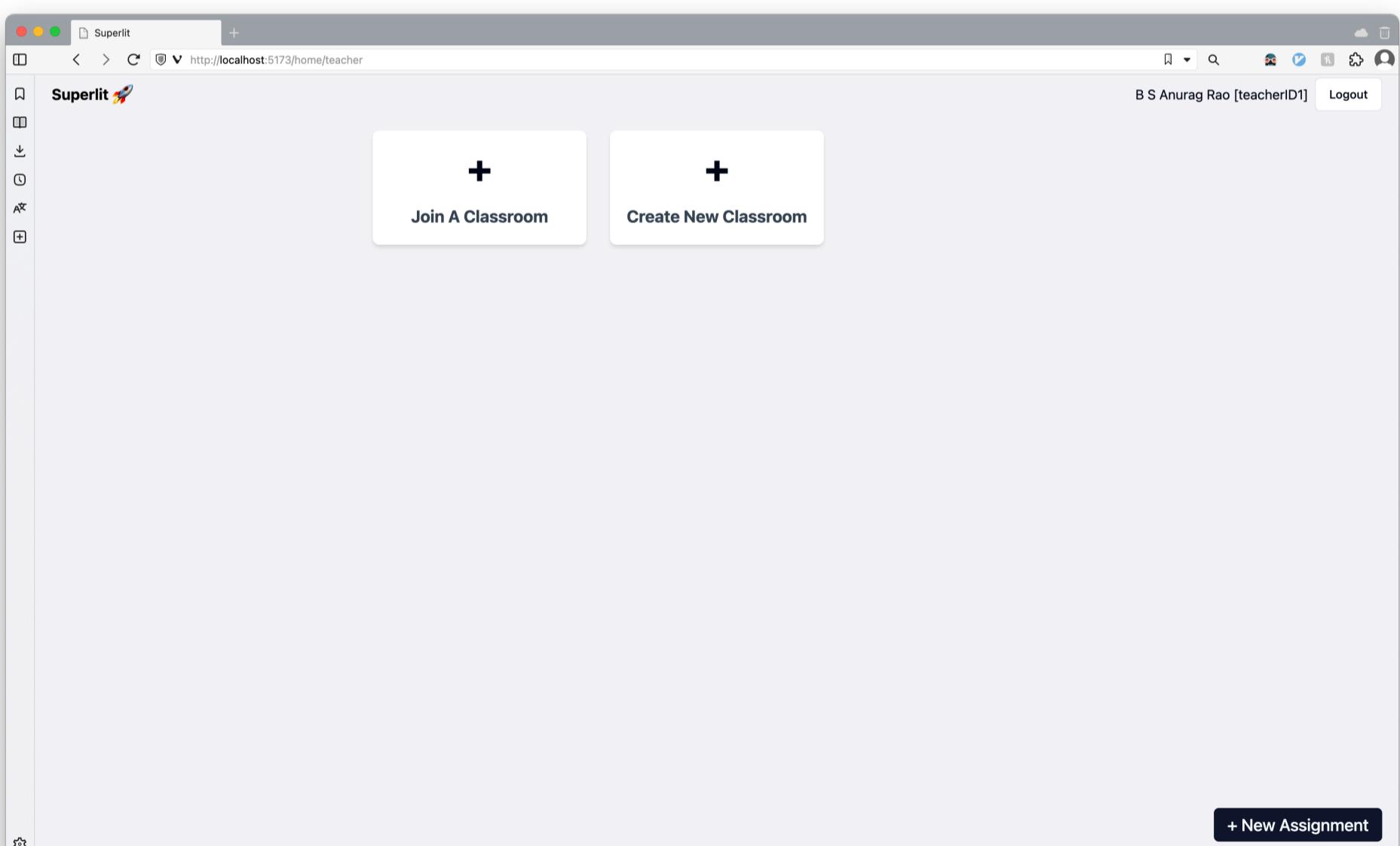
- Here, you can enter your details and create an account. Make sure to use your university provided email ID.
- There is an option to recover your password in case you forget it
- Once you have created an account, you will be redirected to the sign in page. Use the credentials you just created to sign in.

Sign In & Forgot Password

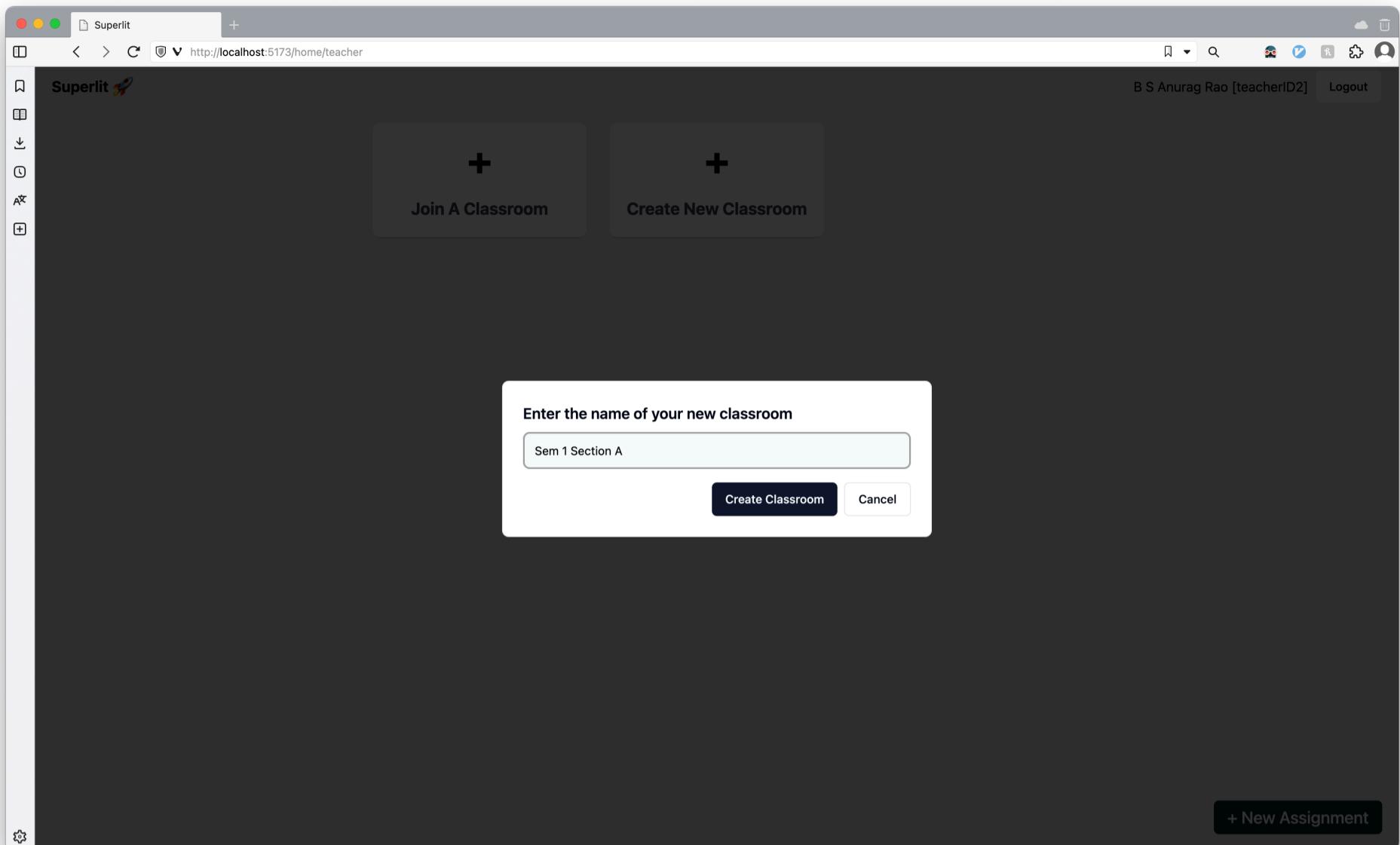
- Use the credentials you just created to sign in. You can use the forgot password option in case you have forgotten your password
- A password reset link will be sent to your registered email ID. You can reset your password by opening that link.



Classrooms



- You will see this interface when you sign in for the first time. The first task to do here is to create a classroom
- It will ask you for the name of the classroom while creating a classroom. This can be something like 'Sem 1 Section A'



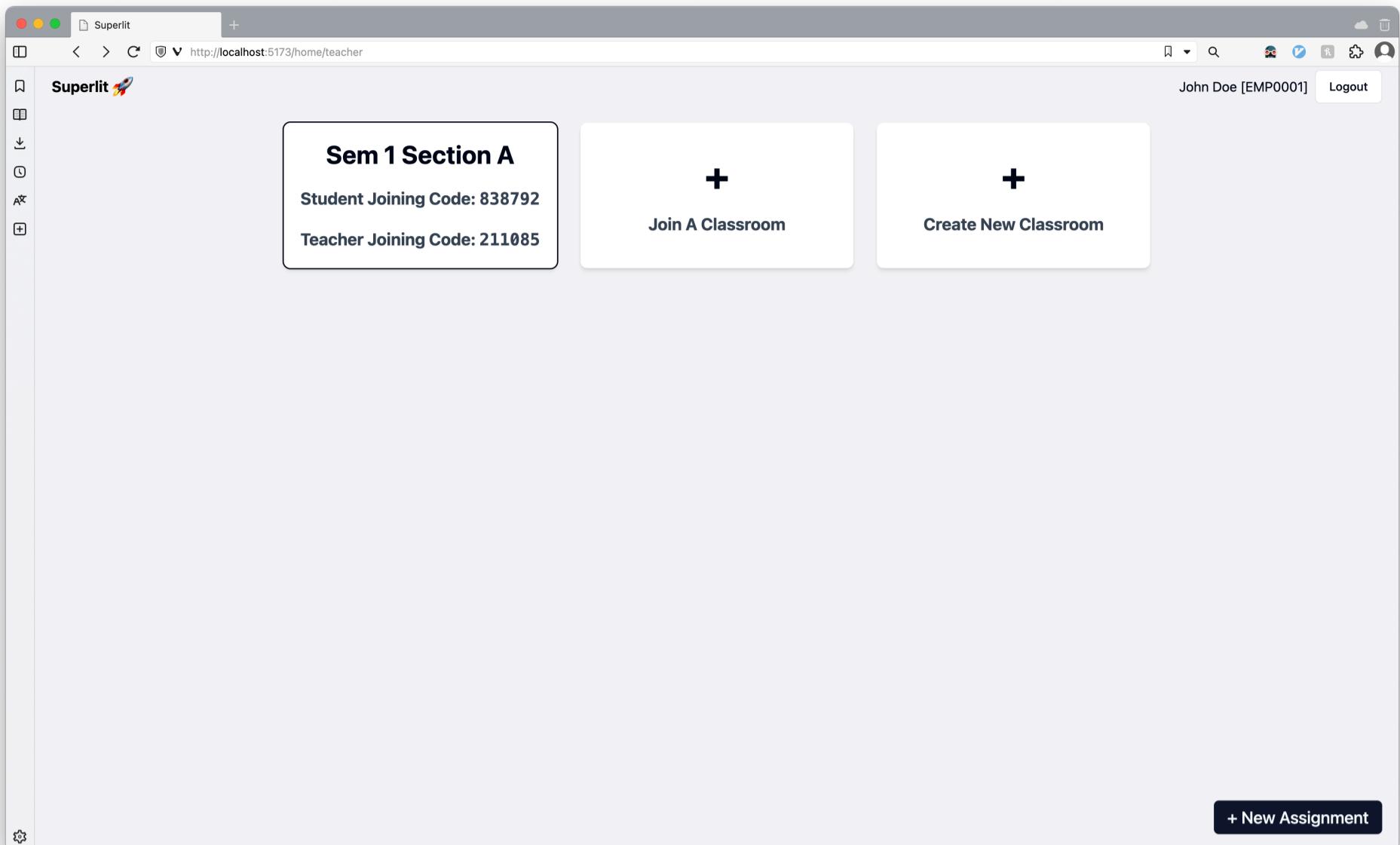
- Once the classroom is created, it will be automatically assigned 2 6-digit codes. One for other teachers to join your classroom and one for other students to join your classroom.
- Share the teacher joining code with other teachers for them to join your classroom.
- Share the student joining code with your class's students for them to join the classroom.

⚠️ Warning

These codes are a secret. Treat it like a password!

DO NOT share the joining codes with anybody than they are intended to be shared with.

More importantly, **DO NOT** share the `teacher joining code` with your students



- If you are a teacher who wants to join another teacher's classroom, then you can use the `join a classroom` card and enter the 6-digit teacher joining code shared to you by the original teacher of the classroom.

Assignments

- Assignments can be tests/assignments you assign to a classroom, or a group of classrooms. To create a new assignment, click on the `+ New Assignment` button on the bottom right of your home page. It will take you to a page that looks like this:

The screenshot shows a 'Create Assignment' form on a web browser. The form has the following fields:

- Name:** A text input field.
- Description:** A text input field.
- Start Time:** A date/time input field with a calendar icon.
- End Time:** A date/time input field with a calendar icon.
- Assign To Classrooms:** A dropdown menu with placeholder text "Select classrooms to assign to..."
- Question 1:** A section header. It contains:
 - Question Title:** A text input field.
 - Question Description:** A text area.
 - Pre-Written Code:** A text area.
 - A red **Delete** button.

- The `Name` and `Description` fields specify the name and description of this assignment. For example the `Name` can be `Lab-1` and `Description` be `First Week - Print Hello World`
- These two are for identifying this assignment, so use names that easily identify this assignment.
- Next, you must set the start and end times of this assignment. Your students will be allowed to submit this assignment only between these times. The start and end time can be across days as well. For example, start today at 10AM and end tomorrow at 11:59PM.
- The `assign to classrooms` field specifies the classrooms you want to assign this assignment to. You can either select only one classroom or select multiple classrooms if you want to assign the same assignment to multiple classrooms. This is useful when you're part of multiple classrooms and there is a common lab session.
- `pre written code` is the code that would be pre written for the students when they answer this question. You can use this field to write the main function and other function signatures and ask the student to only fill in/implement a function.
- `languages allowed` specify the languages the student can write their answer in. If only a single language is selected, then they would be allowed to only answer in that language. If multiple are allowed, the student is given a choice on which language to answer in.
- Next, there are multiple fields for filling in the questions for your coding assignment. The `example cases` are visible as examples to your students. `hidden test cases` are not revealed to the students. The rest of the terms in this section are as per their usual meaning. You can use the `+ Add` buttons to add more example cases, test cases etc.
- Every test case (example and hidden) have their own score. You can use this field to give different weightage to different test cases. An edge case can have a higher score. A student who's code is able to pass the edge case as well will be rewarded in that case. The final score for that question is the sum of scores achieved in all the test cases passed.

The screenshot shows a web-based assignment creation tool. At the top, there's a toolbar with icons for file operations, search, and other common functions. Below the toolbar, a purple button says '+ Add Constraint'. The main area is divided into two sections: 'Example Cases' and 'Hidden Test Cases'.

Example Cases:

- Input:** 2
- Expected Output:** hello world
hello world
- Score:** 2
- Explanation:** 'hello world' is printed twice as the input is 2

+ Add Example Case

Hidden Test Cases:

- Input:** 3
- Expected Output:** hello world
hello world
hello world
- Score:** 2
- Input:** 0
- Expected Output:** (empty)
- Score:** 6

+ Add Test Case

A note on the left side of the 'Hidden Test Cases' section states: "Edge Case has more points than the other cases" with an arrow pointing to the first test case.

AI Constraints

- The most interesting field here is the **AI Constraints** field.
- We use an AI system to verify design constraints and principles in the student's code.
- For example, let's place a constraint in our code that the **student must print** `hello world` '**n**' **times recursively**. This is purely for demonstration. You can have constraints that say `must use structures`, `must use an array of pointers`, `must use a binary tree`, `must use a for loop` etc. This is to enforce that the students use the principles and concepts taught and do not code a solution that does not lead to any learning.

ⓘ Info

You can also enforce time and space complexities by entering an AI constraint that says `must be an O(logn) solution`

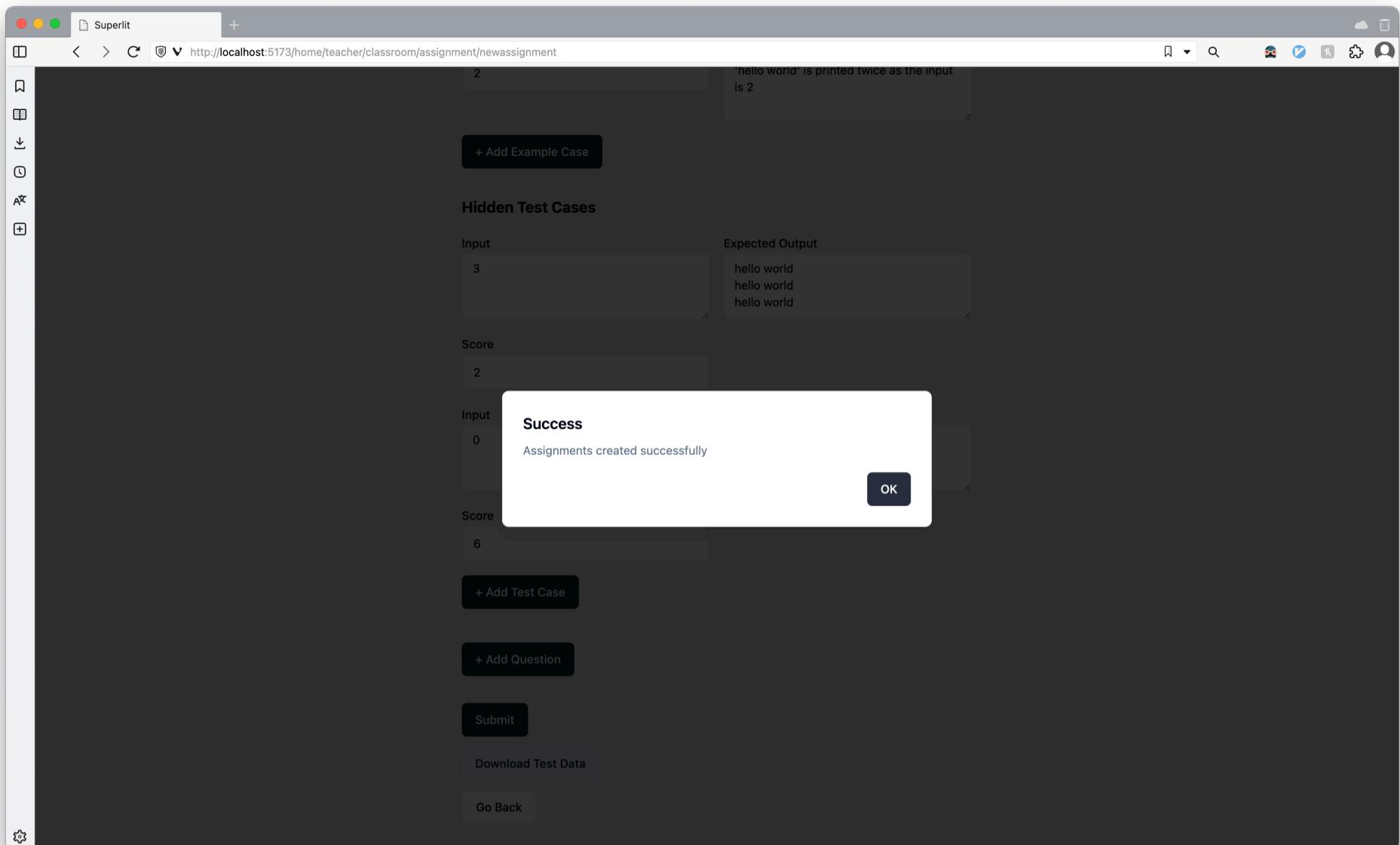
- So, we create 1 constraint that specifies `must use recursion to print hello world`

The screenshot shows a web-based application for creating assignments. At the top, there's a header bar with icons for file operations, a search bar, and user account information. Below the header, the main content area is titled "Assign To Classrooms" with a dropdown menu showing "Sem 1 Section A". There's a red "Delete" button next to it. The next section is "Question 1", which includes a "Question Title" field containing "Hello World" and a "Delete" button. Below that is a "Question Description" field with the instruction: "Print 'hello world' n times recursively. Take the input number 'n' from the standard input." Underneath is a "Pre-Written Code" field, which is currently empty. The next section is "Languages Allowed", showing a dropdown menu with "C" selected. Below that is the "AI Constraints" section, which contains a single constraint: "must use recursion to print hello world n times". There's a blue "+ Add Constraint" button below this. The final section shown is "Example Cases", which includes "Input" and "Expected Output" fields, both of which are currently empty.

- Make sure you mention these constraints in the question description as well. These AI Constraints are **not** shown to the students. They only see the question title and question description
- These constraints are later verified when the student submits their code. If the student has not followed these constraints, they are flagged and you will be able to view their code to evaluate further and take any action you find appropriate.

Download Test Data

- You can download the details of this test in a `.json` file to your own computer
- You can later upload the same data using the `upload test data` button at the top right corner of the page. You can use this option if this is a critical high stakes test like an ISA. The test data is not uploaded to the server unless you hit `submit`.



List Assignments

- You can click on a particular classroom to view all the assignments assigned to that particular classroom. These are sorted in the order they are assigned.

Sem 1 Section A					
Assignments					
Name	Description	Start Time	End Time	View Scores	Edit
Lab-1	First Week - Print Hello World	1/8/2024, 12:00:00 pm	2/8/2024, 11:59:00 pm	View Scores	Edit

- From here, you can edit your assignment (change the test cases, add/delete questions, etc) by clicking on the `Edit` button next to the assignment you want to edit.
- You can `View Score` by clicking on the view score button

View Score

- Once students have joined your classroom, they will be able to see and attempt all the assignments posted to that classroom.
- After they submit, a table of their score breakdown is displayed on this page
- For this example, one student has submitted:

The screenshot shows a web browser window titled "Superlit" with the URL <http://localhost:5173/home/teacher/classroom/887506/assignment/2/scores>. The page displays a table titled "Student Scores" with one row of data. The columns are: University ID, Question 1, Question 1 AI Verify, Question 1 AI Viva Score, Question 1 Student's Code, and Total Score. The data row shows: studentID, 10, Check Student's Code! Why?, 4/4, Download, and 10. Below the table are three buttons: Download as CSV, Populate Google Sheet, and Verify With AI.

University ID	Question 1	Question 1 AI Verify	Question 1 AI Viva Score	Question 1 Student's Code	Total Score
studentID	10	Check Student's Code! Why?	4/4	Download	10

[Download as CSV](#) [Populate Google Sheet](#) [Verify With AI](#)

AI Verification

- This field specifies if the student's code follows the `AI Constraints` set by you earlier while setting the question
- As we can see, in this example, the AI Verification asks you to check the student's code.
- You can get to know why by hovering on the blue tooltip `Why?`:

The screenshot shows a web browser window titled "Superlit" with the URL <http://localhost:5173/home/teacher/classroom/887506/assignment/2/scores>. The page displays "Student Scores" for a student with University ID "studentID". The student has a score of 10 for Question 1, which was checked by AI and got 4/4. A message indicates that the code does not use recursion to print 'hello world' n times as required by the constraint. It uses a for loop instead. There are buttons for "Download", "Download as CSV", "Populate Google Sheet", and "Verify With AI".

- Now you can see what the student has written as an answer by clicking on the `Download` button for that question. This will download the student's code to your computer and you can open it using your favourite text editor (VS Code, Notepad, Vim etc)

The screenshot shows the same Superlit interface as above, but with a download dialog box overlaid. The dialog is titled "Download 'studentID_02/08/2024.txt'?". It shows the file name is "studentID_02/08/2024.txt", file size is unknown, host is Data URL, and location is /Users/anuragrao/Downloads. There is a checkbox for "Always Save Files to Default Download Location". Below the dialog are buttons for "Save", "Save As...", "Open", and "Cancel". The main "Student Scores" table on the right shows the student's code for Question 1 and a total score of 10.

University ID	Question 1	Question 1 AI Verify	Question 1 AI Viva Score	Question 1 Student's Code	Total Score
studentID	10	Check Student's Code! Why?	4/4	Download	10

[Download as CSV](#) [Populate Google Sheet](#) [Verify With AI](#)

```
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    for(int i = 0; i < n; i++){
        printf("hello world\n");
    }
}
```

Student's Code.
As we can see, No recursion is used and hence, our AI system flagged it

- If the student is found to really have been not following the design principles, you can call the student and ask him to submit again.
- The student will also be able to see the AI Verification status on their dashboard. So they can independently decide to attempt again as well given that the assignment deadline is not over.
- You can also re run the AI Verification by clicking on the [Verify With AI](#) button.

AI Viva

- This is another novel AI Feature built into superlit.
- After every submission, the student is asked 4 questions generated based on their own code. The questions are generated **highly specific** to the submitted code.
- The score of this viva is displayed on the table above and is also visible to the student
- You can use this score to determine if the student truly understands their own code or has probably copied off of somewhere else.

The screenshot shows a web application window titled "Superlit". The main content area is titled "Student Scores" and displays a table of student scores. The columns are: University ID, Question 1, Question 1 AI Verify, Question 1 AI Viva Score, Question 1 Student's Code, and Total Score. A row for "studentID" shows a score of 10, with "Question 1 AI Verify" showing "4/4" in green. Below the table are three buttons: "Download as CSV", "Populate Google Sheet", and "Verify With AI". An arrow points from the text "Check Student's Code! Why?" to the "Verify With AI" button. The status bar at the bottom of the browser shows "Downloading Update".

Download as CSV & Populate Google Sheet

- You can download this table as CSV to make changes and for further processing of scores.
- If you want this table in a google sheet, you can create a google sheet and either:
 1. enable write permission for the google account sheets@superlit.iam.gserviceaccount.com
 2. or enable write permission for everybody temporarily

The screenshot shows a Google Sheets interface with a sharing dialog box open over a spreadsheet. The dialog is titled "Share 'Superlit Testing'" and contains fields for "Add people, groups, and calendar events" and a "Copy link" button. To the right of the dialog is a sidebar titled "Tables" with various pre-built table templates. The main spreadsheet has a single row of data starting with "Type '@' then a file name to insert a file smart chip". The status bar at the bottom of the browser shows "Downloading Update".

← Share "Superlit Testing"



sheets@superlit.iam.gserviceaccount.com



Editor ▾



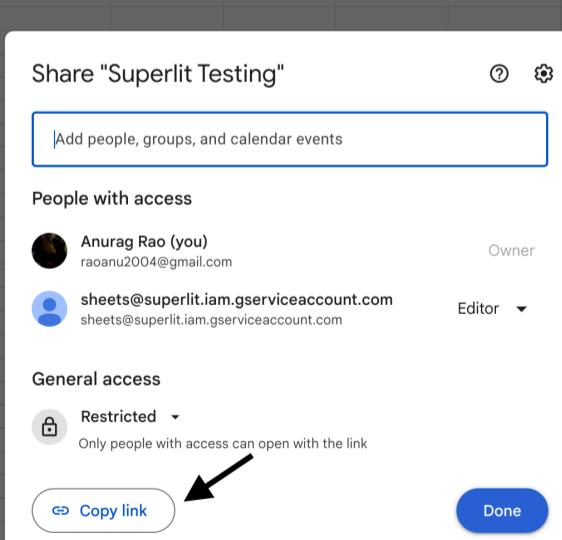
Notify people

Message



Cancel

Send



A screenshot of a Google Sheets spreadsheet titled "Superlit Testing". The sharing dialog is open, showing the email "sheets@superlit.iam.gserviceaccount.com" with access level "Editor". The "Copy link" button is highlighted with a black arrow.

Superlit Testing - Google Sheets

File Edit View Insert Format Data Tools Extensions Help

Access updated

Share

Tables

Start with pre-built tables

Featured

- Event tasks
- Project tasks
- Content tracker
- Product prioritization
- Recruitment applicants

Event planning

Customer relations

Project management

Share "Superlit Testing"

Add people, groups, and calendar events

People with access

Anurag Rao (you)
raoanu2004@gmail.com Owner

sheets@superlit.iam.gserviceaccount.com
sheets@superlit.iam.gserviceaccount.com Editor

General access

Restricted

Only people with access can open with the link

Done

Copy link

Sheet1

- Then copy the share link and paste it into the superlit platform:

The screenshot shows a web browser window for the Superlit platform. The main page displays student scores for a classroom assignment. A modal window is open in the center, titled "Enter Your Google Sheet Link". It contains instructions to enable edit access for the account "sheets@superlit.iam.gserviceaccount.com" or enable edit access for everybody temporarily. Below the instructions is a text input field containing a Google Sheets URL: "1kP3Z-TZKIM0_Dq5LFcyq3I7ubgfyCtMkx107qduc/edit?usp=sharing". At the bottom of the modal are two buttons: "Populate Sheet" (in white text on a dark background) and "Cancel".

- Next, click the `Populate Sheet` button. This will create a new sheet in your google sheet and populate the scores there.

✓ Note

The list of 'sheets' in a given google sheet is displayed on the bottom tab bar in the google sheets website. You will need to find the new sheet created here

Superlit Testing - Google Sheets

Type "@" then a name to insert a people smart chip

Newly Created Sheet Is Here

Sheet1 Sheet2

Tables

Start with pre-built tables

Includes smart chips, placeholders, and custom color palettes

Featured

Event tasks

Project tasks

Content tracker

Product prioritization

Recruitment applicants

Event planning

Customer relations

Project management

This screenshot shows a Google Sheets document titled 'Superlit Testing'. The first sheet, 'Sheet1', contains a single cell with the text 'Type "@" then a name to insert a people smart chip'. A large black arrow points from the text 'Newly Created Sheet Is Here' to the bottom of the sheet area, indicating where a new sheet was inserted. The second sheet, 'Sheet2', is currently selected and is empty. On the right side of the interface, there is a sidebar titled 'Tables' which lists various pre-built table templates.

Superlit Testing - Google Sheets

University ID

studentID

Question 1 Score

Question 1 Average Total Score

Check Student's

10

Sheet1 Sheet2

Tables

Start with pre-built tables

Includes smart chips, placeholders, and custom color palettes

Featured

Event tasks

Project tasks

Content tracker

Product prioritization

Recruitment applicants

Event planning

Customer relations

Project management

This screenshot shows a Google Sheets document titled 'Superlit Testing'. The first sheet, 'Sheet1', contains a table with three rows of data. The columns are labeled 'University ID', 'Question 1 Score', 'Question 1 Average Total Score', and 'Check Student's'. The first row has the value 'studentID' in the 'University ID' column and '10' in the 'Check Student's' column. The second row has the value 'Question 1 Score' in the 'University ID' column and 'Question 1 Average Total Score' in the 'Check Student's' column. The third row has the value '10' in both the 'University ID' and 'Check Student's' columns. The 'Check Student's' column is formatted as a dropdown menu. On the right side of the interface, there is a sidebar titled 'Tables' which lists various pre-built table templates.