

Final Submission

Group 4

Milestone 1

In the first Milestone the task was to identify the users and the types of users for this application of Placement Portal. The work done is as below. Additionally user stories were also asked to be made and the work is as below.

Task 1 : Identify users of the application - primary, secondary and tertiary users.

The Primary User for the Placement Portal i.e. the most frequent users shall be students who are preparing for their placements or Seniors who have gone through the placements and would like to share their experience, tips, suggestions, questions asked, assessments,projects.

The secondary users for the Placement Portal are those users who don't use the Portal directly but use it through an intermediary. The Placement Team Members, IITM BS PODheads, HOD, Director, IIT Senate could go through the briefs of the Placement Portal from quarter to quarter to note down the feedback of students as to how they are faring in placements so that based on that the standard of the program can be varied, what all improvements can be carried out to ease the placement process, which all skill sets are expected by companies so that those can be included in the curriculum, what type of projects are expected by students so they can be included in the degree. A sentiment analysis also could be carried out to give a much better idea.

The tertiary users are those users that don't use the software at all but are affected by the use of the Software in our case the Placement Portal. These users in our case are the various companies that are coming to the IIT M BS Program to directly recruit students as interns or as employees. Based on the experience, tips, suggestions, assessments and resources shared by Seniors to students sitting for placements the performance of these students shall be correlated. And the better the inputs most likely better is the performance of students and higher rate of placements. Placement Preparation websites such as Leetcode , Hackerank, Codechef, Kaggle ,etc shall also become a tertiary user as these companies shall be affected heavily by the introduction of the Portal. These preparation websites offer various services and resources to help prepare students and Seniors could point out which ones are of importance to the students hence. In addition, if the Portal has its own assessment system then the traffic on these system could reduce at least for the customer base of those users coming from IITM BS degree.

Task 2 : Write user stories for the requirements, based on the SMART guidelines discussed in the lectures: The user stories should be in the following format:

As a [type of user], I want [an action], So that [a benefit/value].

User Story 1:

As a student Preparing for Placements I want to be able to post questions as a new thread or in an existing thread regarding placements so that I can learn, correct ,improve and implement from the experiences shared by Seniors who have gone through the process before or are working.

User Story 2:

As a Senior Student who would like to share their experiences, I would want to post answers to questions in threads or create a new one so that I could answer questions to what my juniors have posed as well as give my inputs, experiences, expectations and tips as a whole in a completely new thread.

User Story 3:

As a Junior as well as a Senior Student I would like to have the feature of posting questions as well as answers in an anonymous manner if required so that I would want not to be bothered due to privacy or security concerns while posting questions regarding placements as well as sharing experiences.

User Story 4:

As a Placement Team Member/POD I would want to have a feature to check daily, weekly, monthly stats such as most Active Thread, most Posts liked , most Posts commented, Interview Experiences of Company ABC, TRE, XYZ indexed together ,all suggestions of that duration indexed together so that we are well involved as well as informed of what the current status is of students in their preparations, which all improvements are to be put up, what the expectations of companies are through the experiences shared so that required changes could be made to make students better prepared.

User Story 5:

As a Junior Student/Senior/Placement Council I would want to search threads on the bases of the title, tags, author so that I can view, track and add my inputs on those threads.

Milestone 2

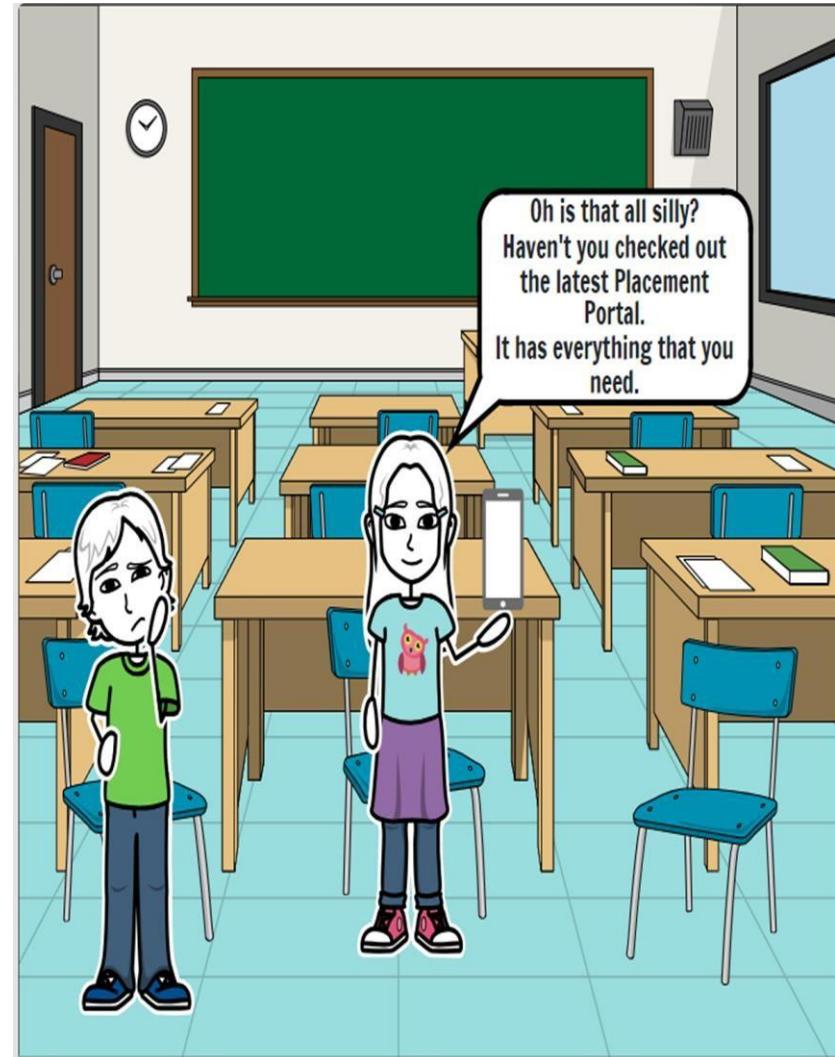
In the second Milestone we were asked to create story board and create a wireframe as well for it. The work done with respect to Milestone 2 is as below.

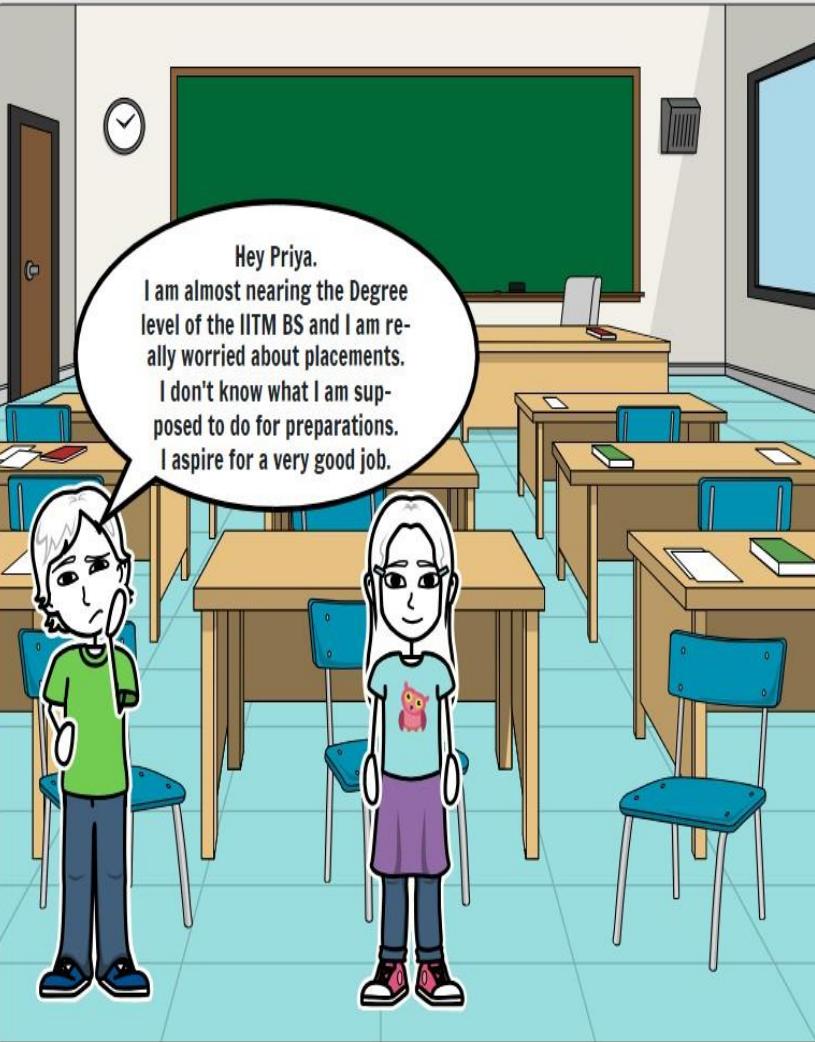
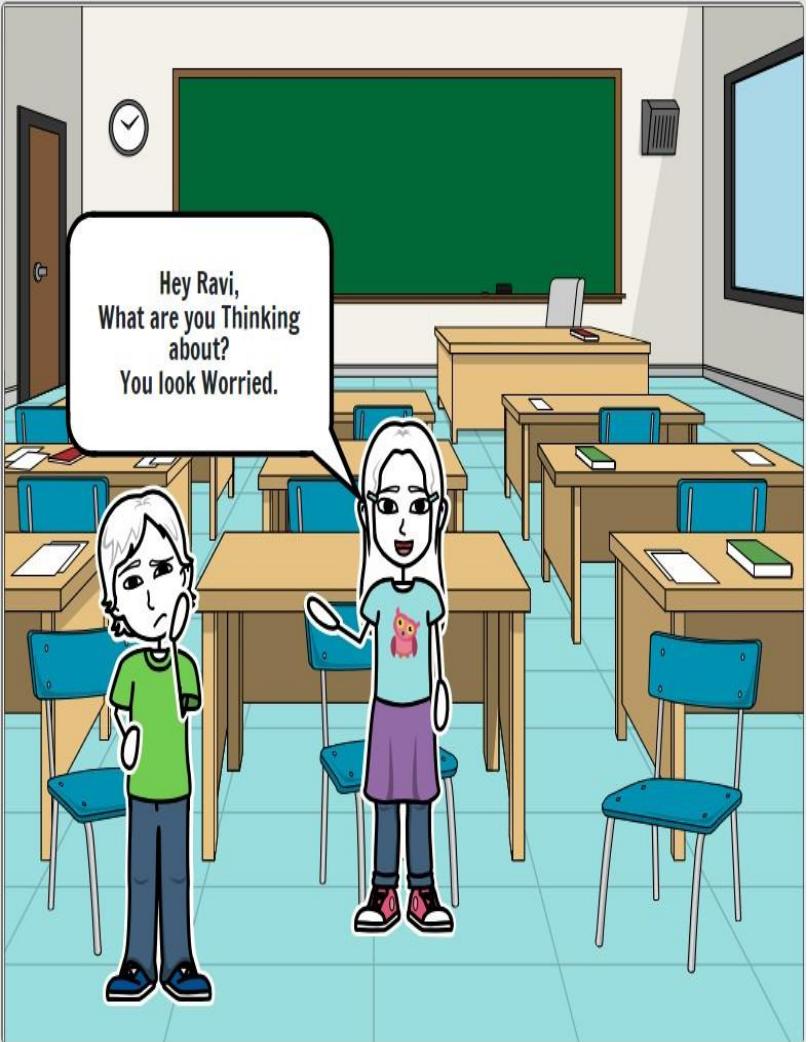
Milestone 2

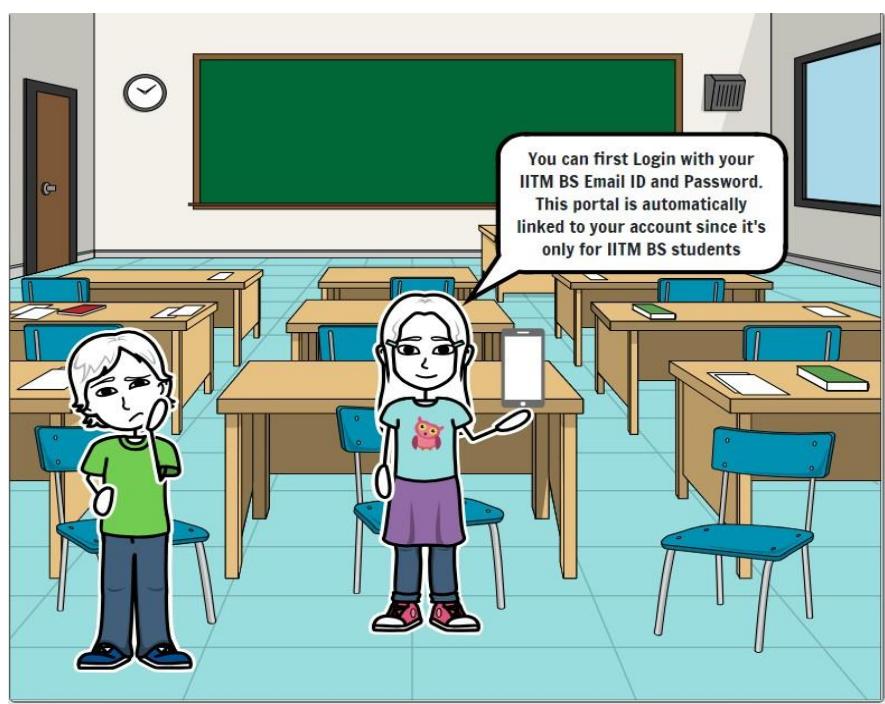
By

Group 4

For Junior Students







For All Users

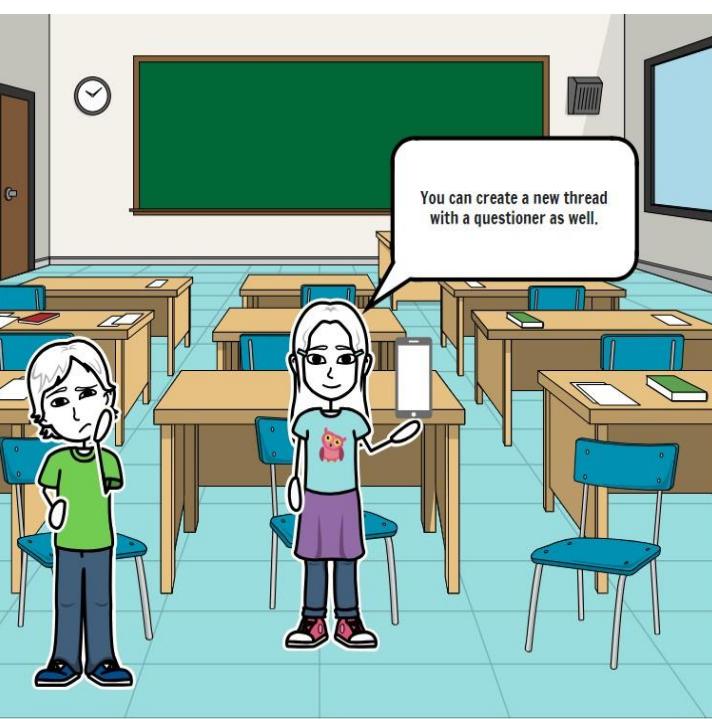
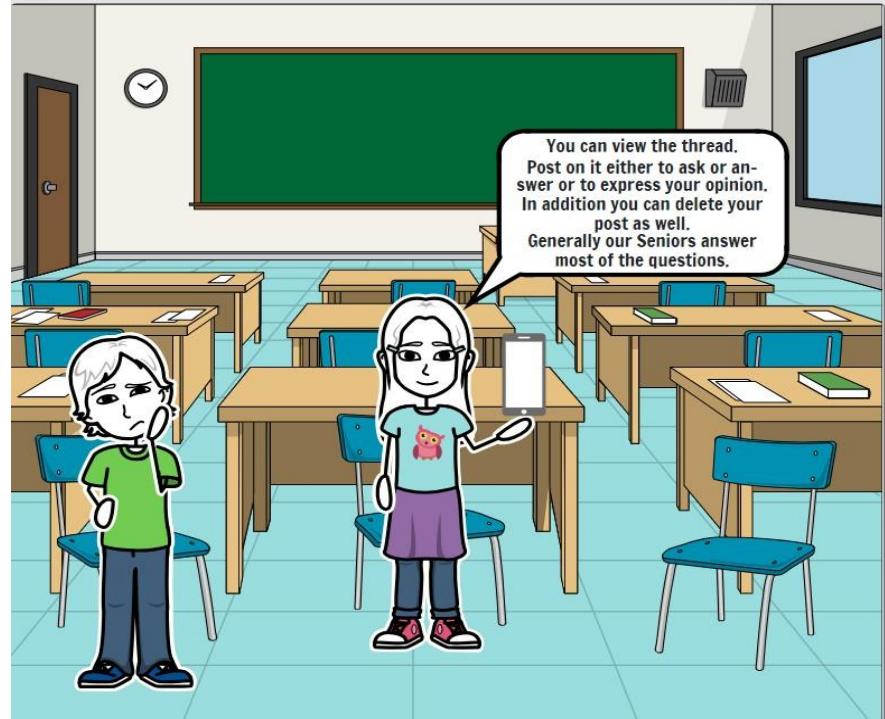
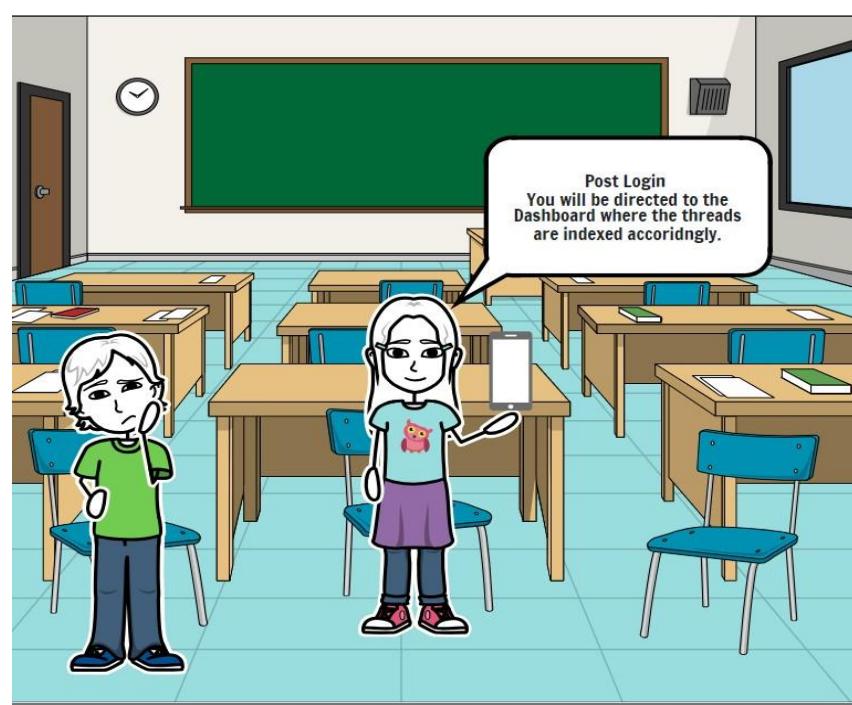
Android Small - 1

Email Address

Password

[Forgot Password](#)

[Reset Password](#)



For Junior Students

Android Small - 1

Dashboard

My Profile Assessments
Hi Ravi My Threads
Logout

Search Thread/Posts

New Thread

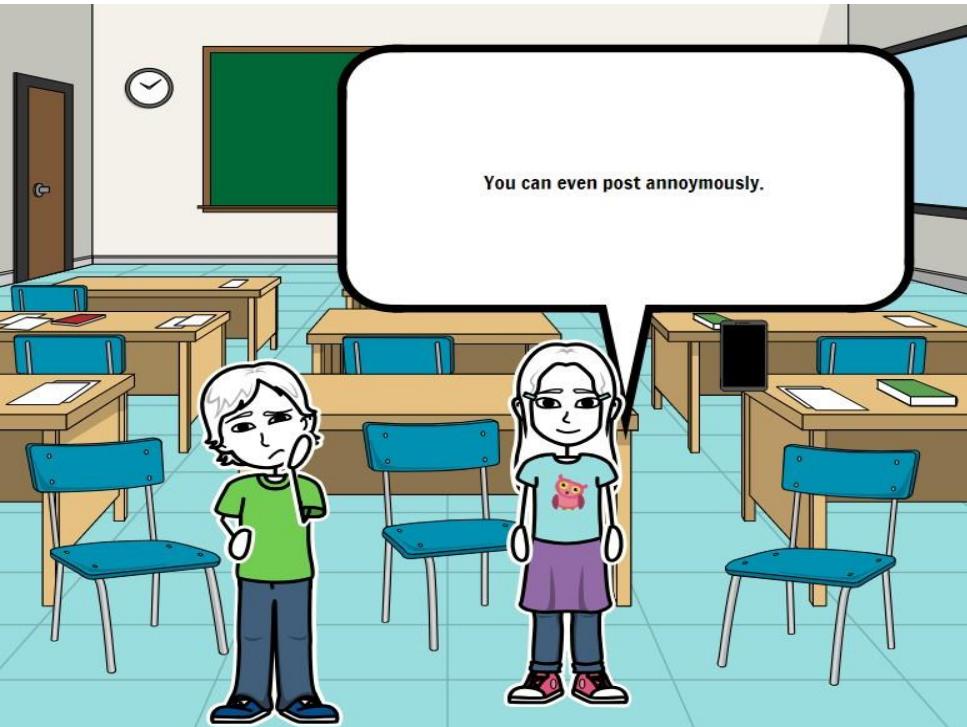
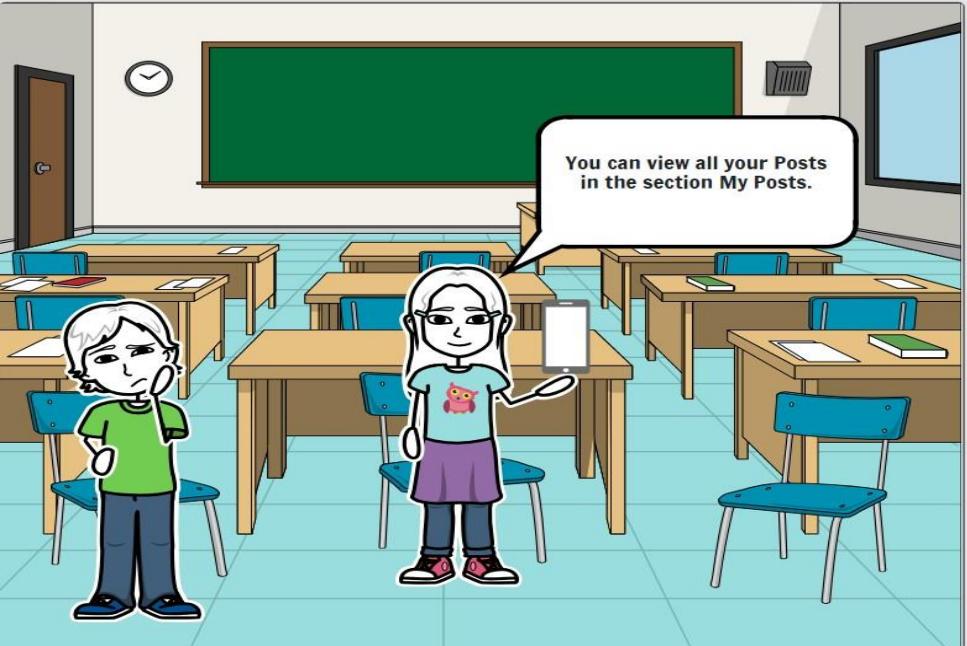
Most Recent Most Commented

Traversing a Linked List 25 Comments
25.04.2020 Thursday 12:01 PM

Time and Distance Problems 12 Comments
23.04.2020 Tuesday 3:07 PM

Facebook Interview Questions 116 Comments
21.04.2020 Sunday 10:07 PM

Linear Regression Questions 251 Comments
21.04.2020 Sunday 4:03 PM



For Juniors

Android Small - 1

Traversing Linked List

Dashboard Assessments Logout

Rohan Monday 2.03.2020 4:00pm

How to traverse a Linked list?

Murli Monday 2.03.2020 5:00pm

Hi Rohan you can need to go through the list by using the next in the node.

Post Comment

Add Post Anonymously

For Junior

Android Small - 1

My Threads

Dashboard Assessments Logout

Most Recent

DSA 101 25.03.2020
Best Algorithms for Search

Machine Learning 101 21.03.2020
Linear Regression

App Developmenet 19.02.2020
Flask and Vue JS

For Junior Students

Android Small - 1

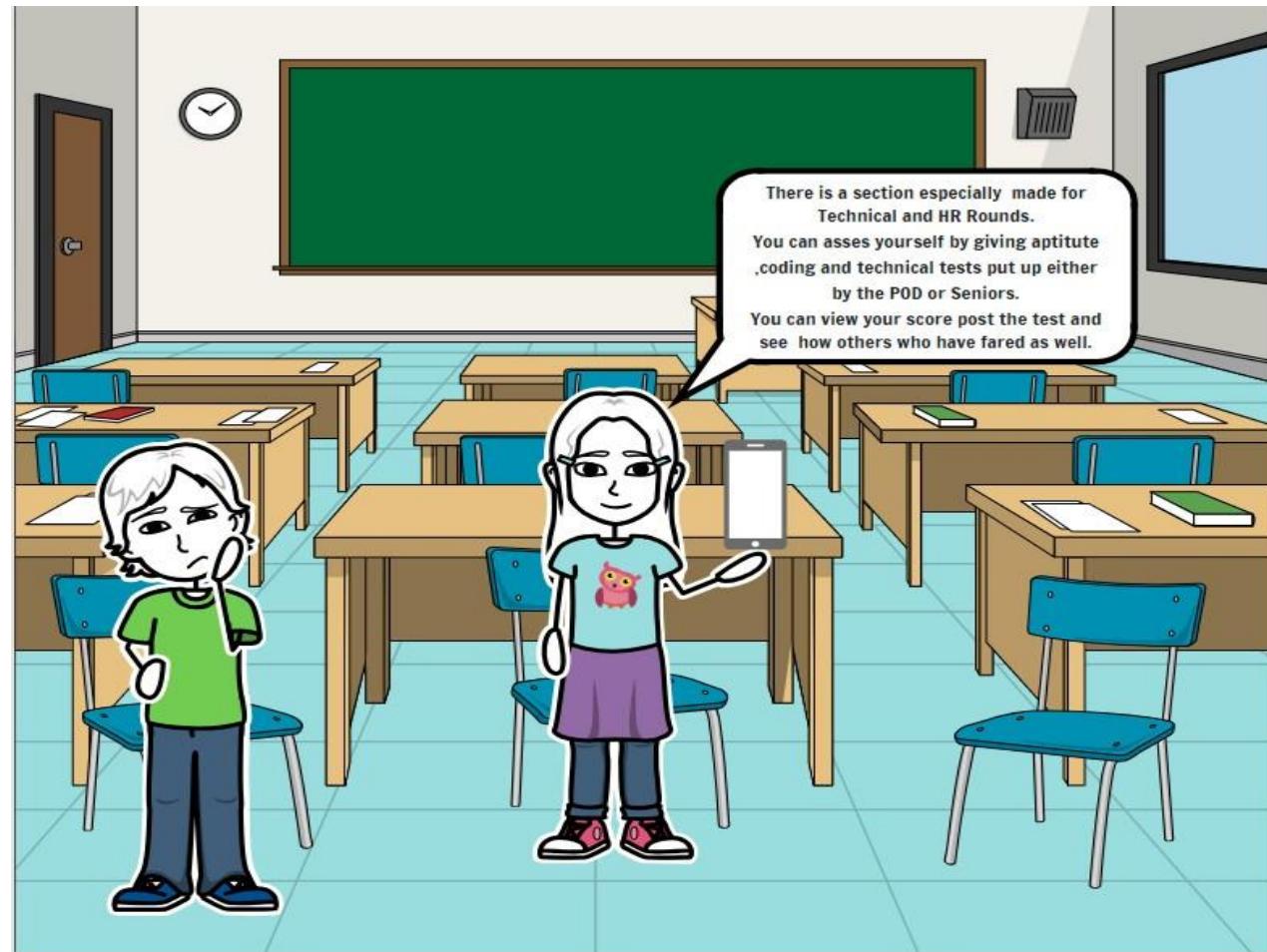
New Thread

Dashboard Assessments Logout

Title

Content

Submit



For Junior

Android Small - 1

Assessments

Dashboard Logout

Data Structures Beginner Quiz

Toppers	Score
Rahul	94
Priya	82
Kashyap	74

[Attempt Quiz](#)

Machine Learning Coding Round

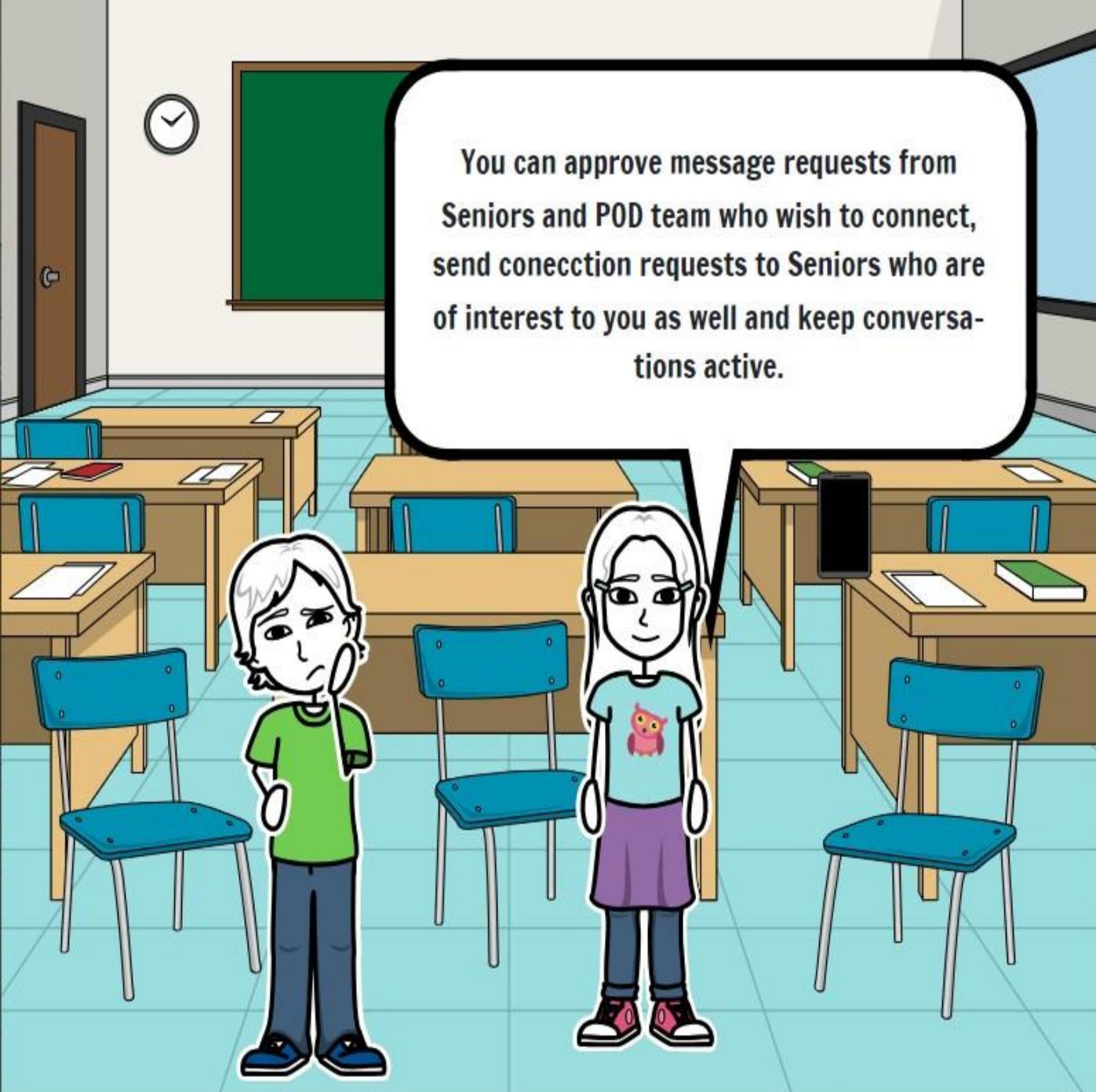
Toppers	Score
Sejal	45
Tripti	42
Gillprit	31

[Check Score](#)

Accenture Aptitude Questions

Toppers	Score
Shreya	91
Shruthi	75
Sruthi	74

[Attempt Quiz](#)



You can approve message requests from Seniors and POD team who wish to connect, send connection requests to Seniors who are of interest to you as well and keep conversations active.

For Junior

Android Small - 1

My Profile

My Profile Dashboard
Assesments
Logout

Name Diya Trivedi
User Type Junior

Send Message Request to User Search Bar

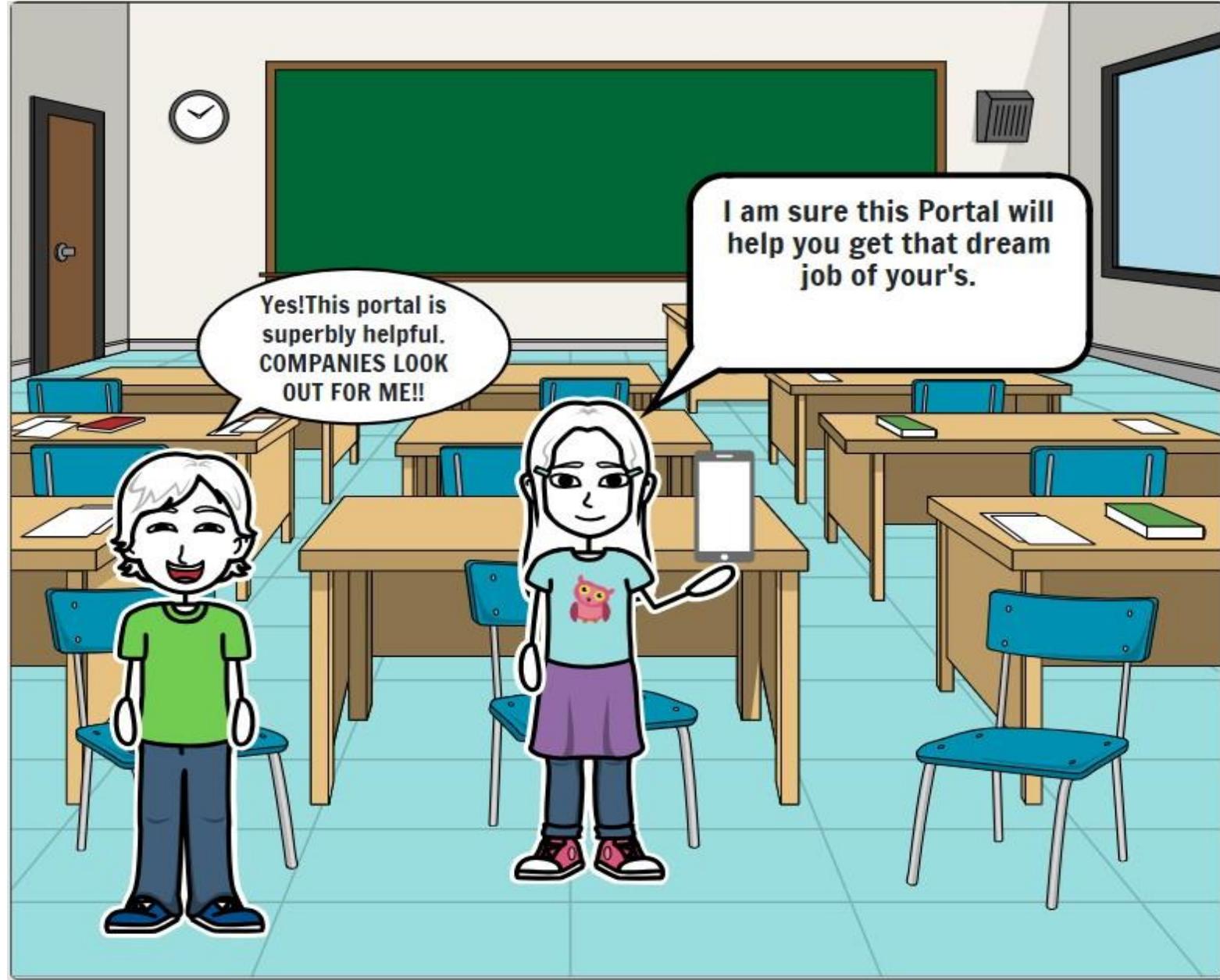
Message Requests

From POD
Arun
Hi,would like to connect for an interview.

From Seniors
Priti
Looking for interns.

Active Messages

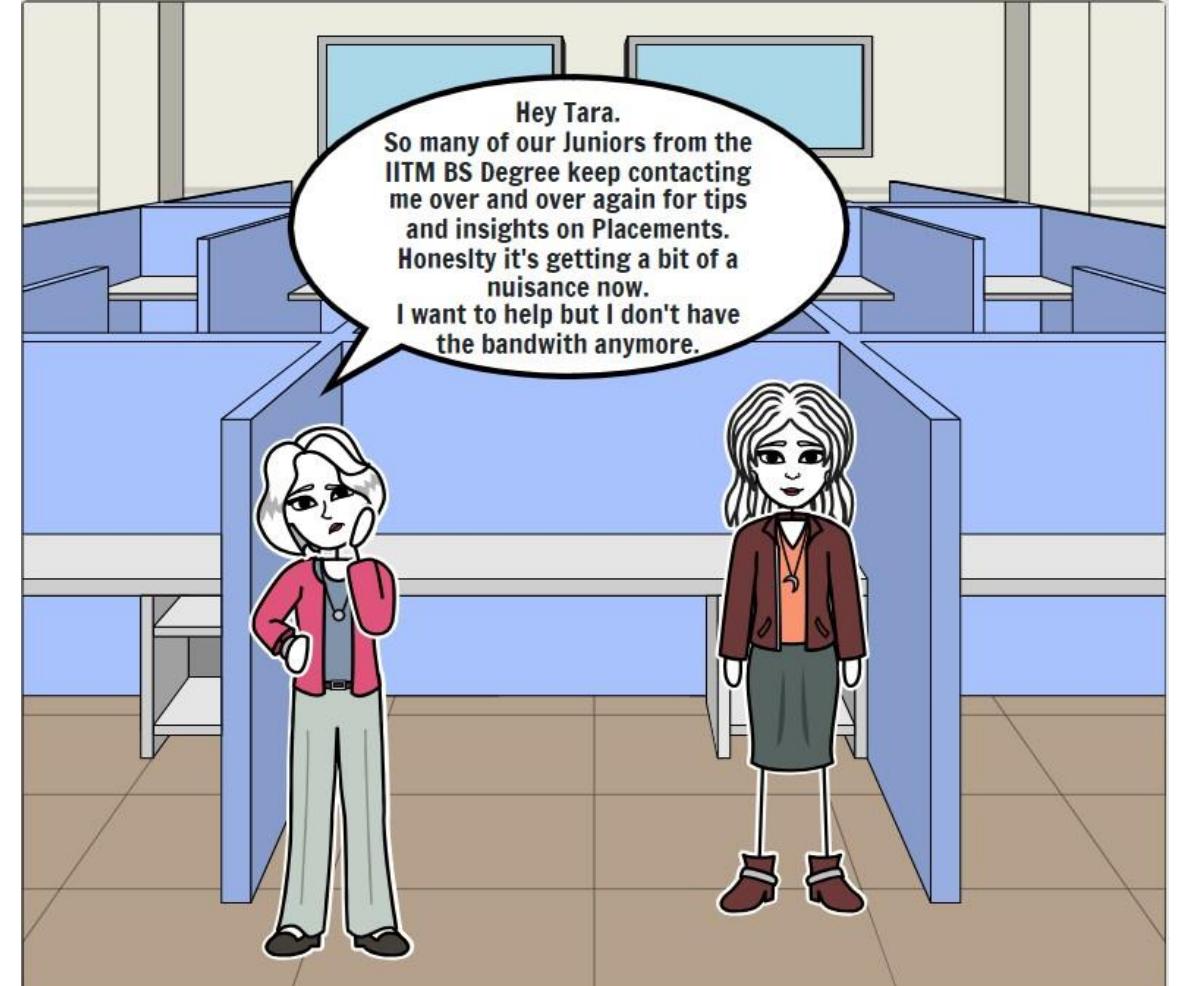
Rahul
Hi Diya,look at this dsa video.



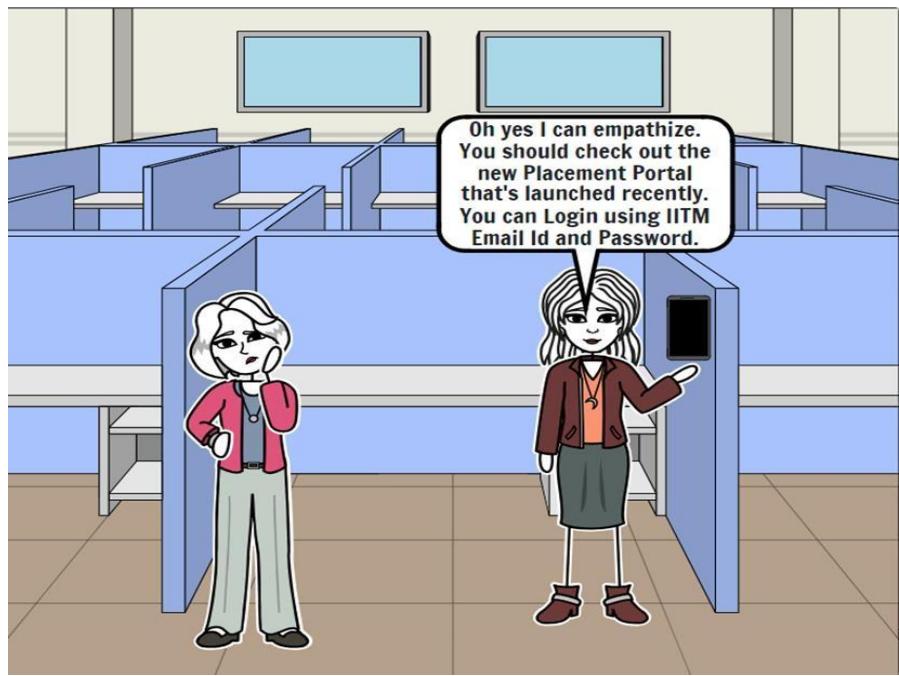
For Senior Students



Hello Diya Ma'am.
How are you doing.
You look really worried.
Is something concerning
you?



Hey Tara.
So many of our Juniors from the
IITM BS Degree keep contacting
me over and over again for tips
and insights on Placements.
Honesty it's getting a bit of a
nuisance now.
I want to help but I don't have
the bandwidth anymore.



For All Users

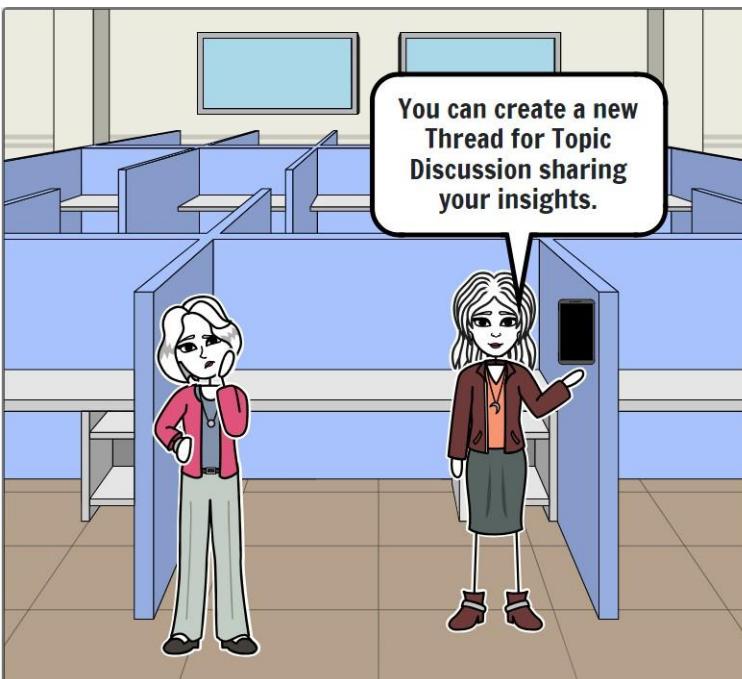
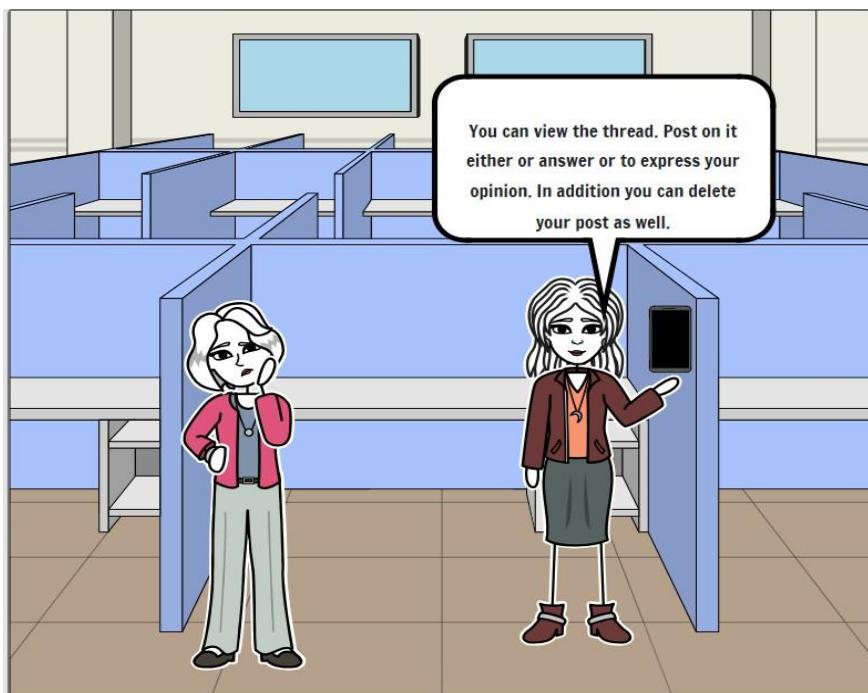
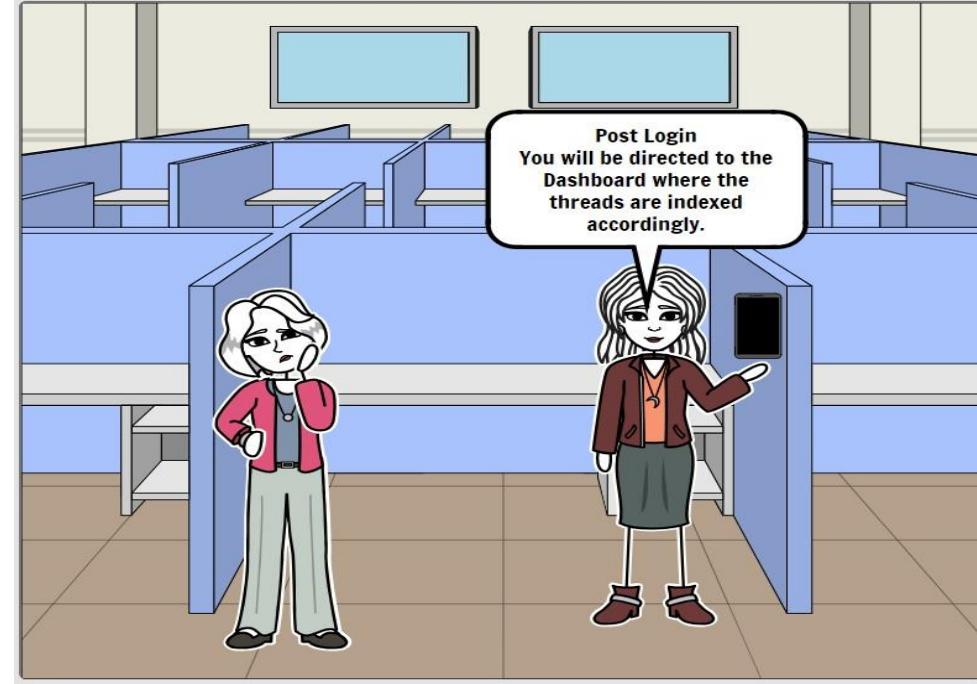
Android Small - 1

Email Address

Password

[Forgot Password](#)

[Reset Password](#)



For Senior Students

Android Small - 1

Dashboard

My Profile My Assessments My Threads Logout

Hi Ravi

Search Thread/Posts

Most Recent Most Commented

Traversing a Linked List 25 Comments 25.04.2020 Thursday 12:01 PM

Time and Distance Problems 12 Comments 23.04.2020 Tuesday 3:07 PM

Facebook Interview Questions 116 Comments 21.04.2020 Sunday 10:07 PM

Linear Regression Questions 251 Comments 21.04.2020 Sunday 4:03 PM



For Juniors

Android Small - 1

Traversing Linked List

Dashboard
Assessments
Logout

Rohan Monday 2.03.2020 4:00pm

How to traverse a Linked list?

Murli Monday 2.03.2020 5:00pm

Hi Rohan you can need to go through the list by using the next in the node.

Post Comment

Add Post Anonymously

For Senior

Android Small - 1

My Threads

Dashboard
My Assessments
Logout

Most Recent

DSA 101 25.03.2020
Best Algorithms for Search

Machine Learning 101 21.03.2020
Linear Regression

App Developmenet 19.02.2020
Flask and Vue JS

For Senior Students

Android Small - 1

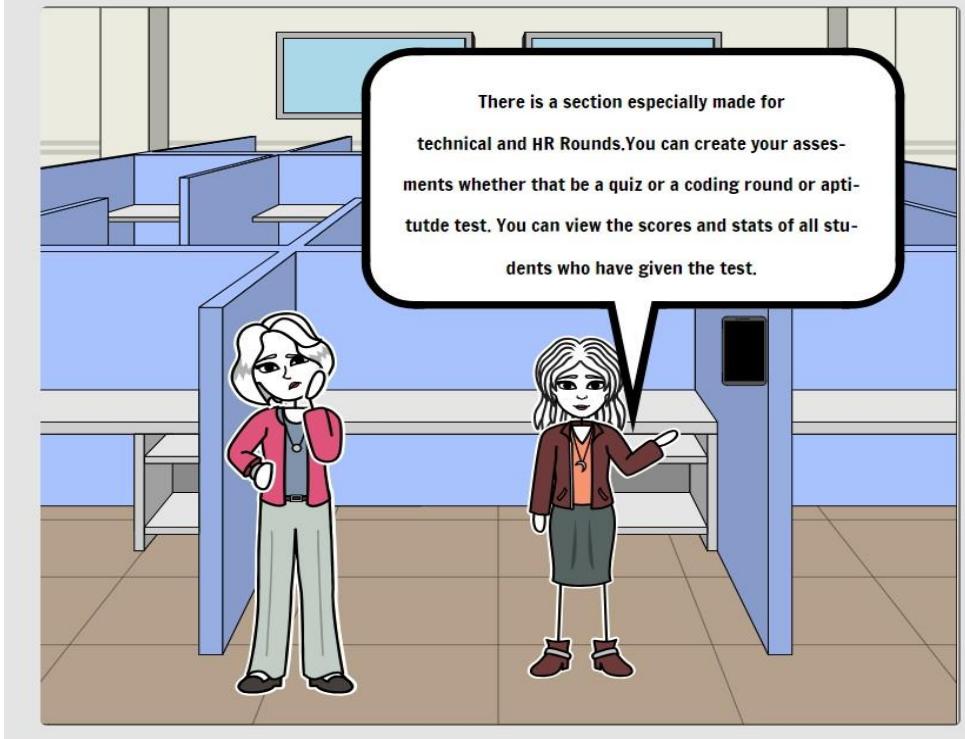
New Thread

Dashboard
My Assessments
Logout

Title

Content

Submit



For Senior

Android Small - 1

My Assessments

Dashboard Logout

Data Structures Beginner Quiz

Toppers	Score
Rahul	94
Priya	82
Kashyap	74

[Check All Score](#) [Check Stats](#)

Machine Learning Coding Round

Toppers	Score
Sejal	45
Tripti	42
Gillprit	31

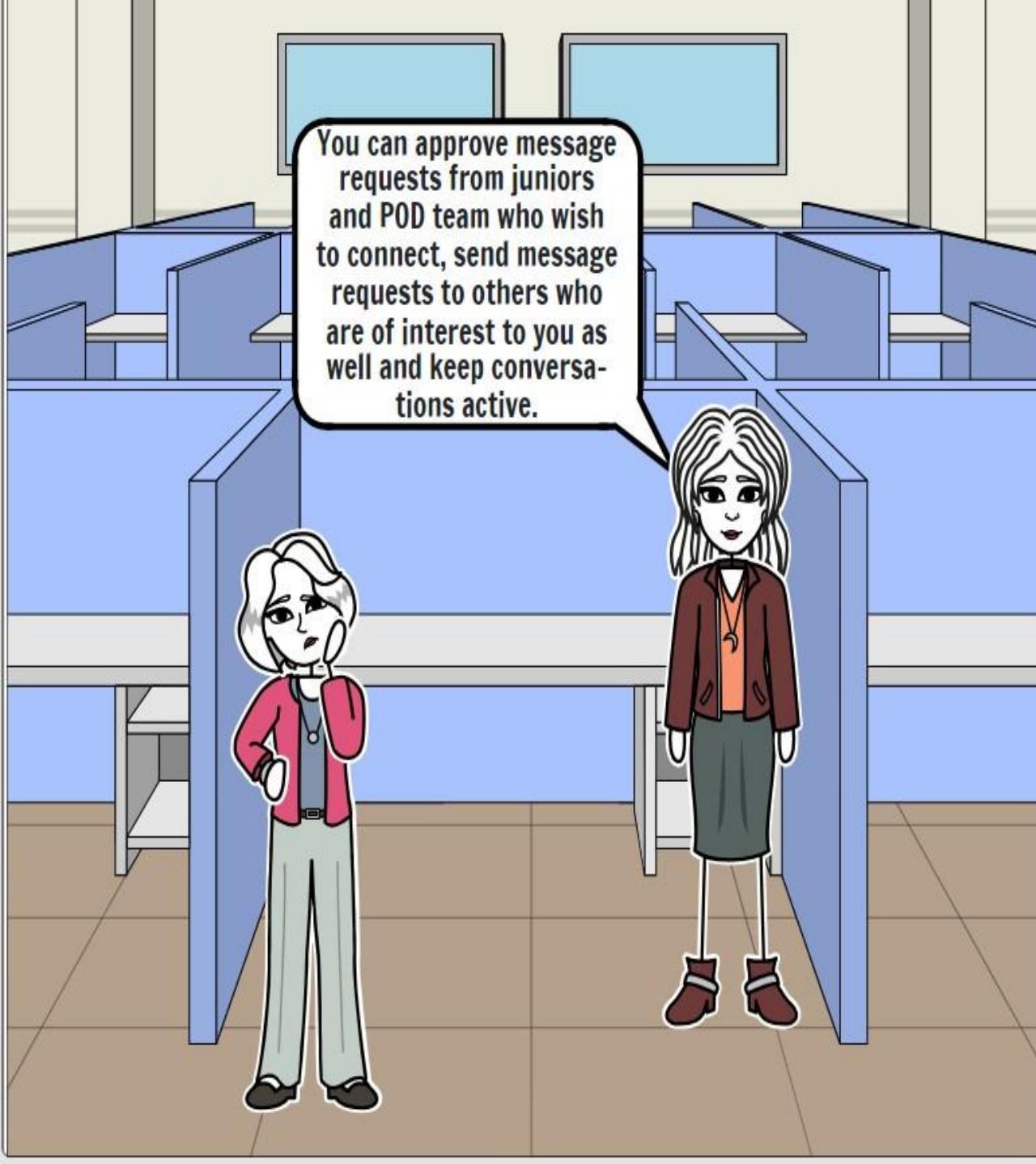
[Check All Score](#) [Check Stats](#)

Accenture Aptitude Questions

Toppers	Score
Shreya	91
Shruthi	75
Sruthi	74

[Check All Score](#) [Check Stats](#)

[Add Assessment](#)



You can approve message requests from juniors and POD team who wish to connect, send message requests to others who are of interest to you as well and keep conversations active.

For Senior

Android Small - 1

My Profile

My Profile Dashboard My Assesments Logout

Name Diya Trivedi User Type Senior

Send Message Request to User Search Bar

Message Requests

From POD

Arun

Hi,would like to connect for an interview.

From Juniors

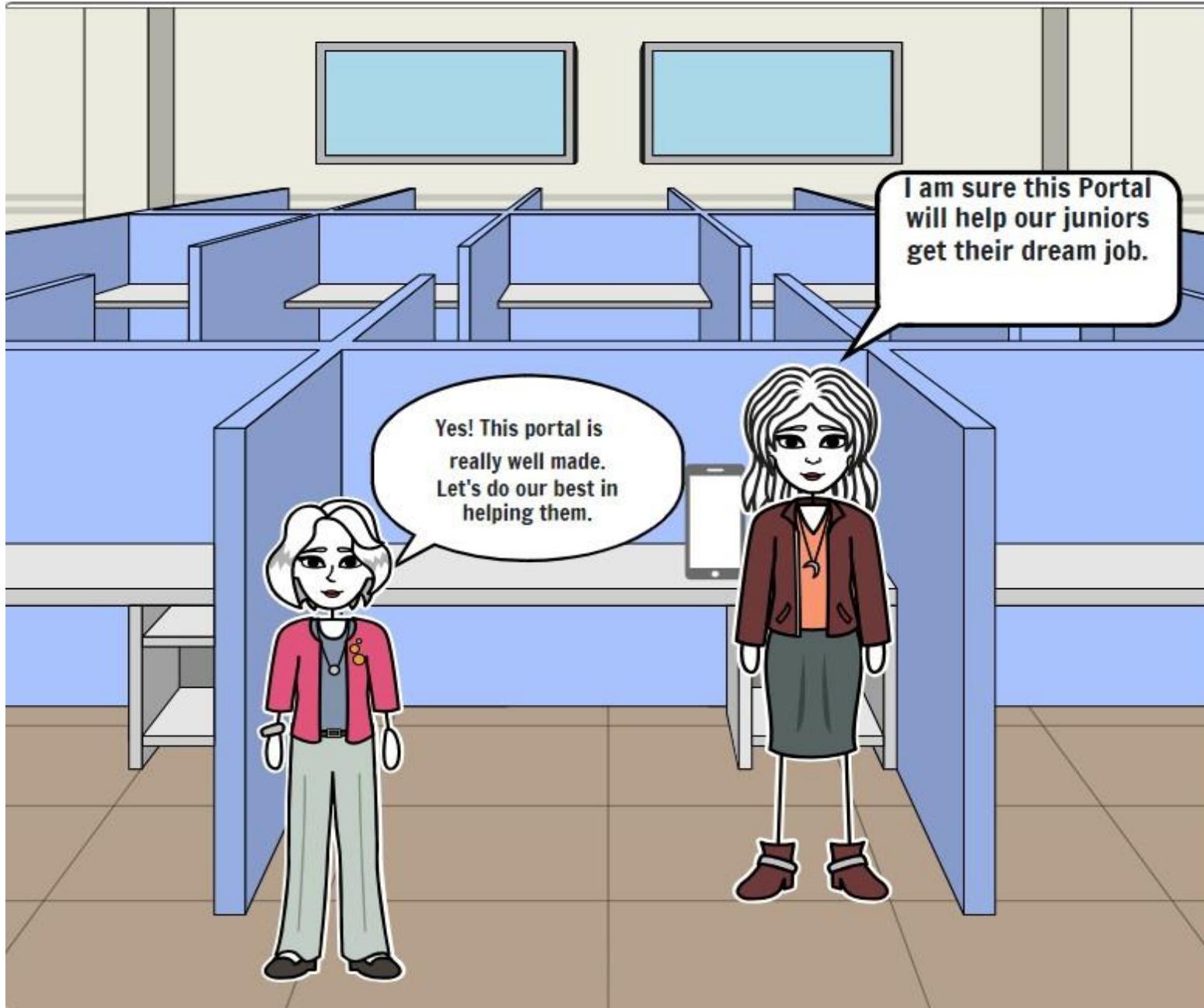
Priti

Need Tips for Interview.

Active Messages

Rahul

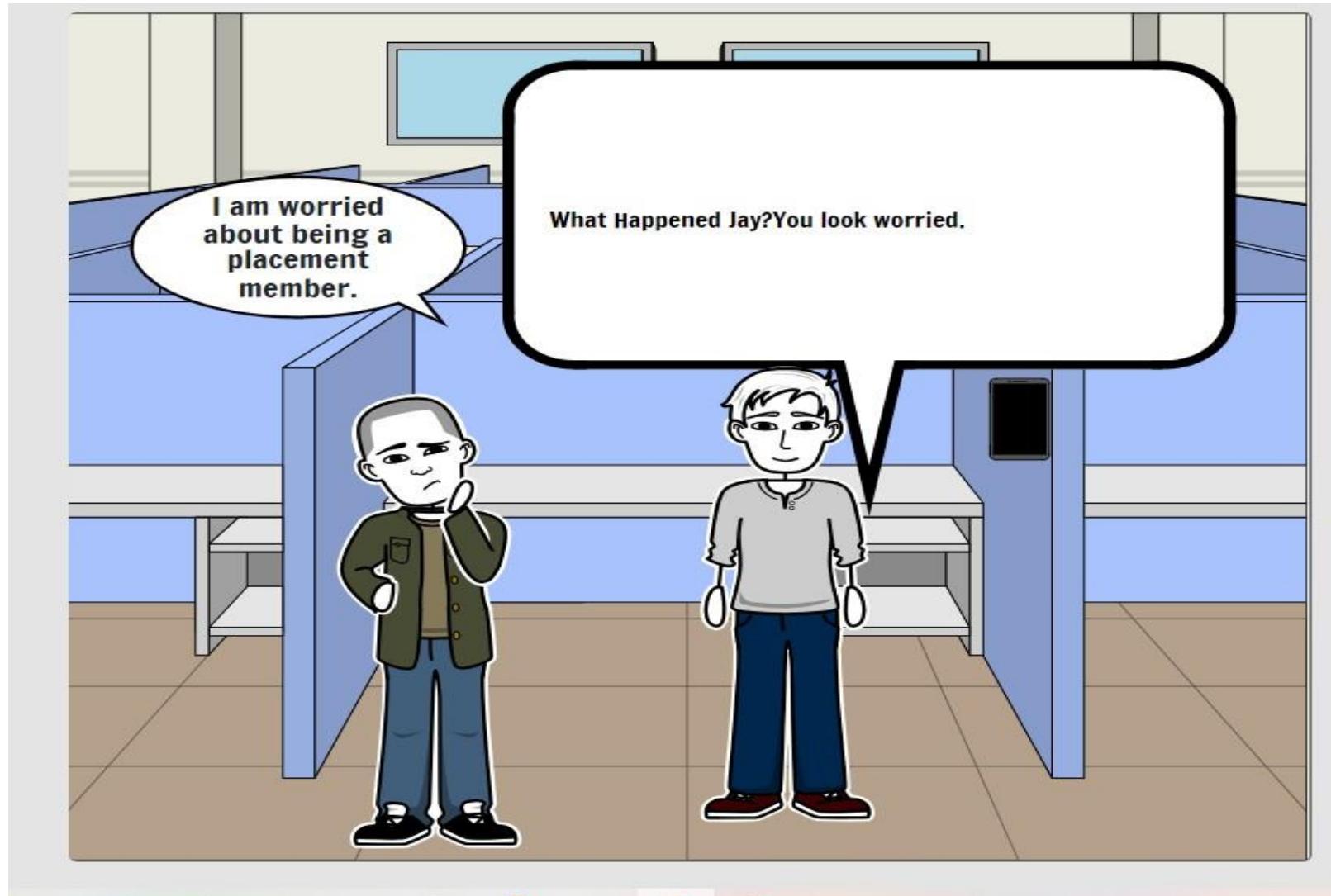
Hi Diya Ma'am.I cracked Google Thank You.

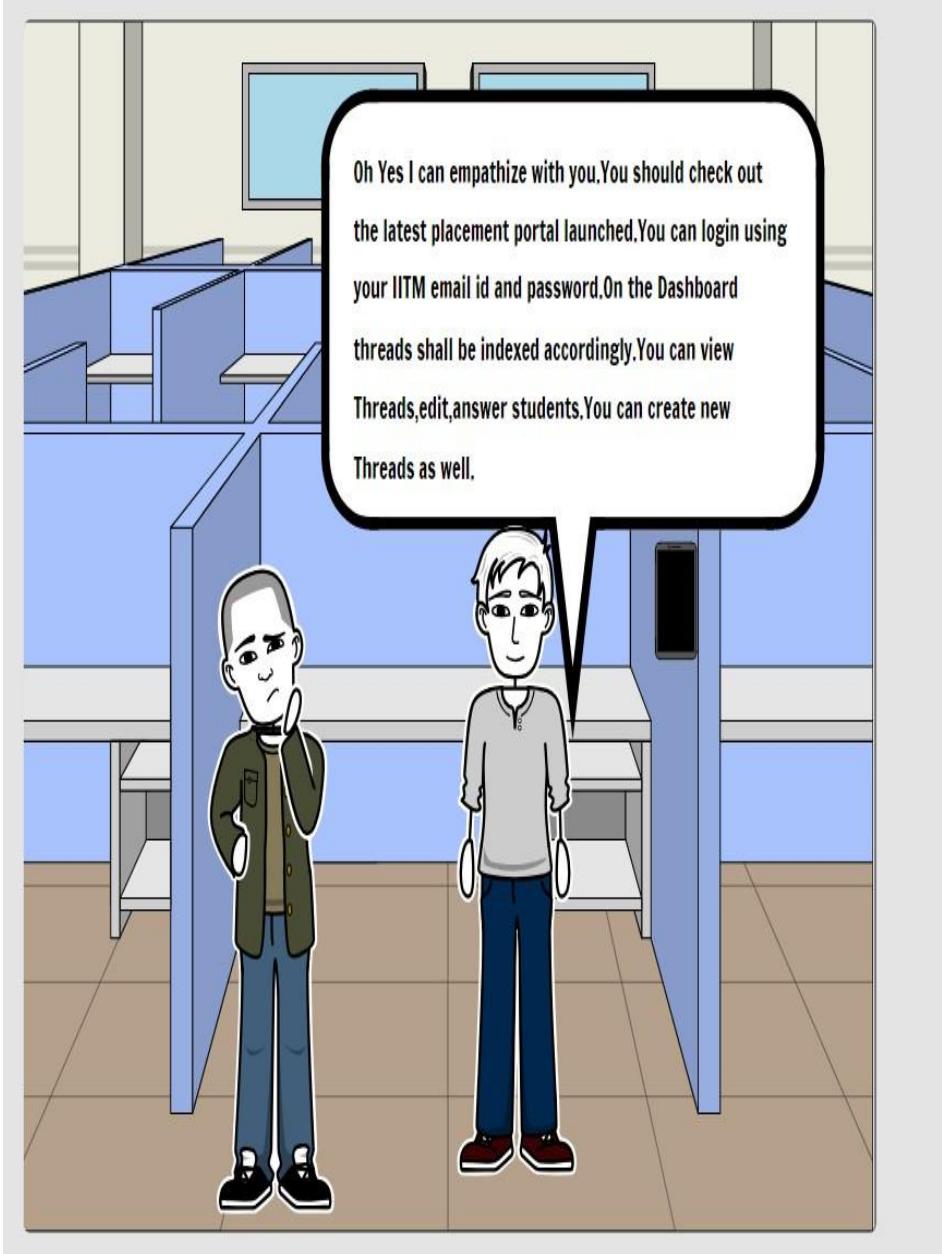


Yes! This portal is
really well made.
Let's do our best in
helping them.

I am sure this Portal
will help our juniors
get their dream job.

For Placement Team





For All Users

Android Small - 1

Email Address

Password

[Forgot Password](#)

[Reset Password](#)

For Placement Team

Android Small - 1

Hi Ravi

Dashboard

My Profile My Assessments
My Threads Stats
Logout

Search Thread/Posts

New Thread

[Most Recent](#) [Most Commented](#)

Traversing a Linked List 25 Comments
25.04.2020 Thursday 12:01 PM

Time and Distance Problems 12 Comments
23.04.2020 Tuesday 3:07 PM

Facebook Interview Questions 116 Comments
21.04.2020 Sunday 10:07 PM

Linear Regression Questions 251 Comments
21.04.2020 Sunday 4:03 PM



For Placement Team

Android Small - 1

My Threads

Dashboard My Assessments Stats Logout

Most Recent

DSA 101 25.03.2020

Best Algorithms for Search

Machine Learning 101 21.03.2020

Linear Regression

App Developmenet 19.02.2020

Flask and Vue JS

For Placement Team

Android Small - 1

Traversing Linked List

Dashboard My Assessments Stats Logout

Rohan Monday 2.03.2020 4:00pm

How to traverse a Linked list?

Murli Monday 2.03.2020 5:00pm

Hi Rohan you can need to go through the list by using the next in the node.

Post Comment

Add

For Placement Team

Android Small - 1

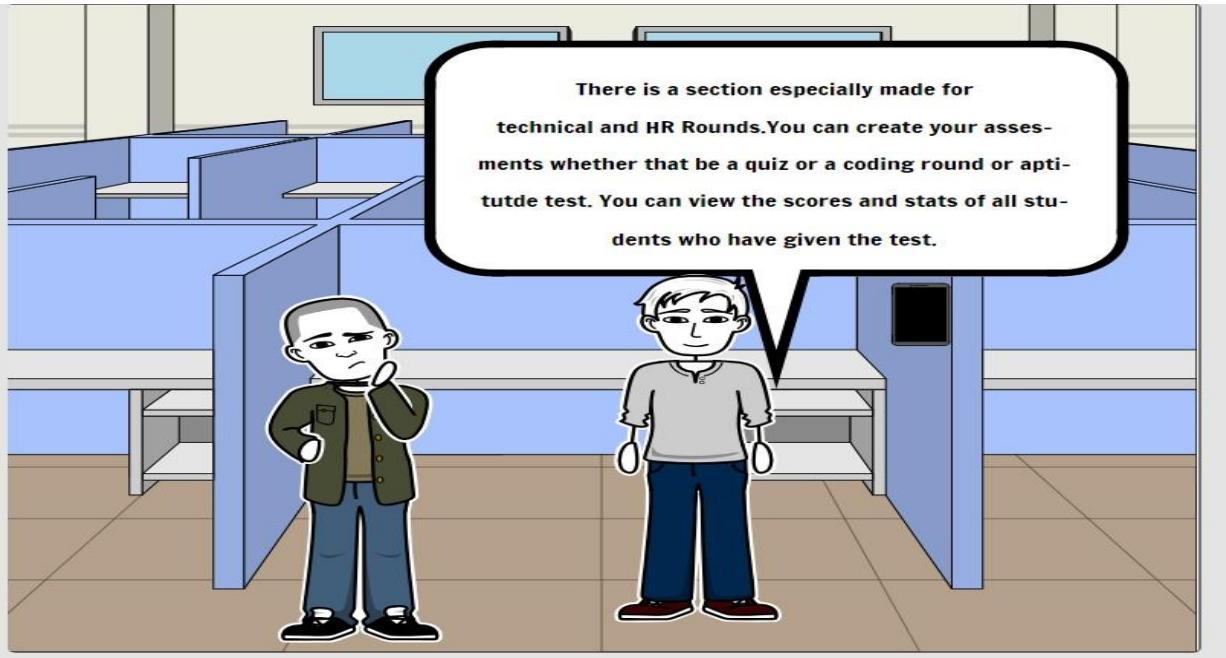
New Thread

Dashboard My Assessments Stats Logout

Title

Content

Submit



There is a section especially made for technical and HR Rounds. You can create your assessments whether that be a quiz or a coding round or aptitude test. You can view the scores and stats of all students who have given the test.



You can even add your own assessments.

The screenshot shows a mobile application interface titled "For Placement Team" with the subtitle "Android Small - 1". At the top right are links for "Dashboard", "Stats", and "Logout". The main content area is titled "My Assessments". It displays three sections: "Data Structures Beginner Quiz", "Machine Learning Coding Round", and "Accenture Aptitude Questions", each showing a list of top scorers with their names and scores, and buttons for "Check All Score" and "Check Stats". At the bottom is a yellow button labeled "Add Assessment".

For Placement Team

Android Small - 1

My Assessments

Dashboard
Stats
Logout

Data Structures Beginner Quiz

Toppers	Score
Rahul	94
Priya	82
Kashyap	74

[Check All Score](#) [Check Stats](#)

Machine Learning Coding Round

Toppers	Score
Sejal	45
Tripti	42
Gillprit	31

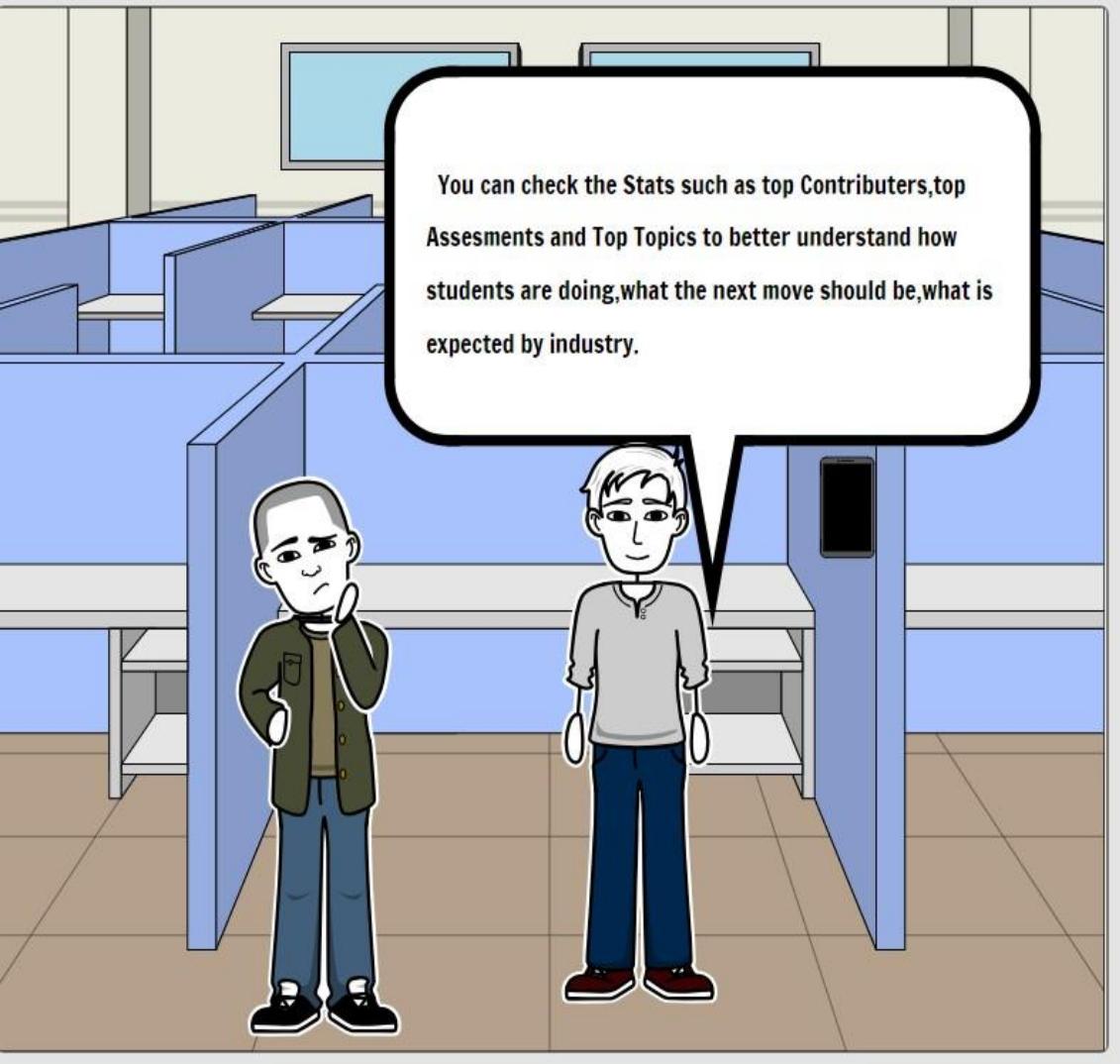
[Check All Score](#) [Check Stats](#)

Accenture Aptitude Questions

Toppers	Score
Shreya	91
Shruthi	75
Sruthi	74

[Check All Score](#) [Check Stats](#)

[Add Assessment](#)



You can check the Stats such as top Contributors,top Assessments and Top Topics to better understand how students are doing,what the next move should be,what is expected by industry.

For Placement Team

Android Small - 1

Stats

Dashboard My Assesments Logout

Top Contributors

Rahul	Posts 94
Priya	82
Kashyap	74

Check List

Top Topics

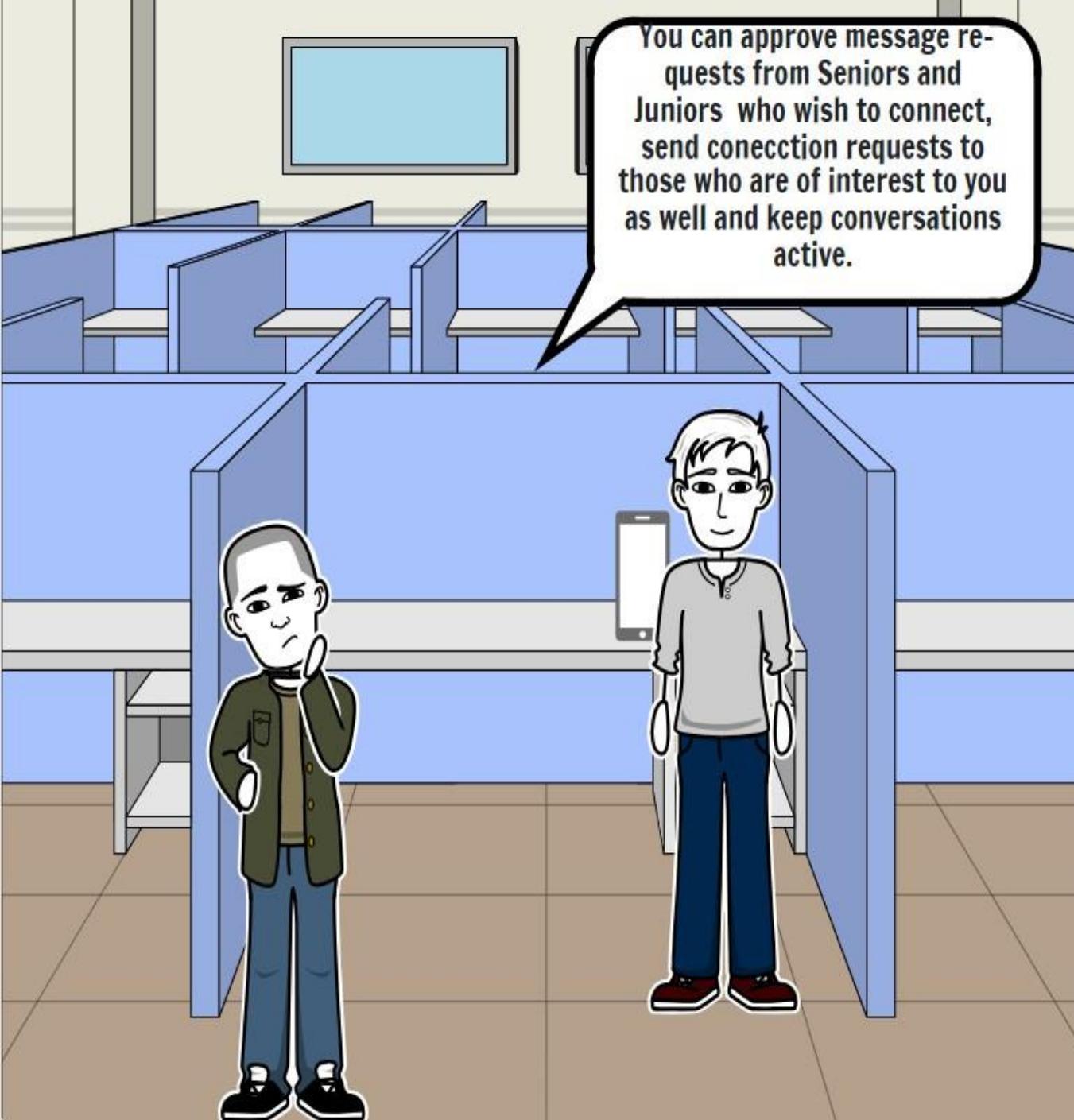
DSA	Threads 45
MLP	42
App Dev	31

Check List

Top Assesments

Aptitude 101	Count 91
Web Dev	75
Machine Learning	74

Check List



For Placement Team

Android Small - 1

My Profile

My Profile

Dashboard
My Assessments
Stats
Logout

Name Diya Trivedi
User Type POD

Send Message Request to User

Message Requests

From Seniors

Arun

Hi,would like to connect for an interview.

From Juniors

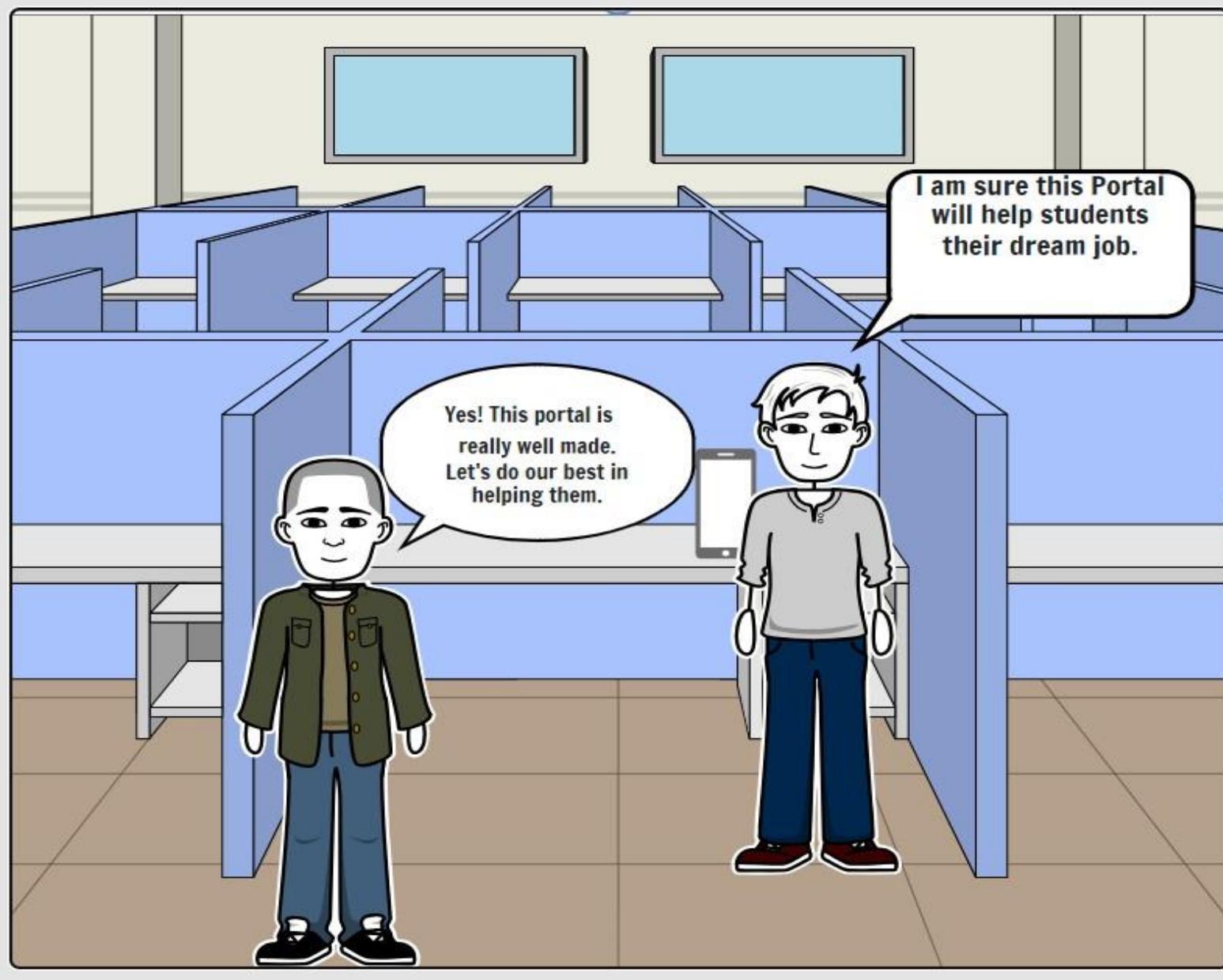
Priti

Looking for internship.

Active Messages

Rahul

Hi Diya,Please have a live session for placement prep.



Milestone 3

In Milestone 3 we were asked to do the Project Scheduling using Jira, design components, implement scrum and take sprint meeting from there on.

The work done is as below.

Project Schedule

Week 1-2: Project Initiation and Planning.

- Understanding the Problem Statement and the types of Users and their Roles,
- Understanding which types of users fall in which bucket i.e Primary, Secondary, Tertiary.
- Gather and prioritize requirements.
- Writing user stories for each Users such as Junior – Primary, Senior- Secondary, POD team – Tertiary.

Week 3: Creating WireFrame and StoryBoard

- Creating storyboard for each User Type using Story Board That.
- Creating a low-fidelity wireframe using Figma.
- Creating a project plan and Schedule for Project Implementation.
- Designing of Components and specifying all the relationships existing between each classes i.e User, Junior, Senior, POD, Threads, Posts, Assessments, Connections, Messages.
- Creating class diagrams for proposed systems.

Week 4: Developing API Endpoints

- Setting up the database for all the classes.
- Develop APIs for user, thread, connection and assessment management.

Week 4: User Management and Authentication

- Implement user registration and login functionality
- Enable user profile creation and management of the Profile

Week 5: Thread and Assessment Management

- Create new thread functionality for posting questions, tips and answers.
- Enable users to post answers to existing threads.
- Enabling Juniors and Seniors to answer Anonymously.
- Enabling Users to search for Threads.
- Enabling Users to view Assessments.
- Enabling Seniors and POD to add Assessments.

Week 6: Connection Management and User Interactions and Engagement

- Final Touching on Assessment Management by enabling Juniors to take quizzes, display their scores.
- Enabling Seniors and POD team to view all scores and stats of Assessments.
- Developing Connection Management System by enabling Users to send messages to others with a message.
- Approval Requests to be maintained.
- Active Conversation Logs to be enabled.
- Implement post liking and commenting functionality.
- Develop a notification system for users to receive updates on threads, messages, requests and assessments.
- Periodic Code reviews.

Week 7: UI/UX Design and Reviewing Code

- Design intuitive and user-friendly navigation
- Periodic Code Reviews done post changes.

Week 8: Testing and Quality Assurance

- Conduct unit testing for individual components
- Perform integration testing for seamless functionality

Week 9: Deployment

- Prepare the production environment for deployment

Week 10: Creating Report

- Creating a detailed report on work done from milestone 1 through milestone 5
- Provide training to placement team members and administrators

Design of Components

1. User View (general foll users):

- Users (junior, senior, and POD team) all will be able to register themselves into the portal using iitm email id only.
- Users need to fill in their full name.
- Accounts are password protected. User need to make a 8 digit password (atleast one uppercase, one lowercase, one number and one special character)
- Users would be registering themselves as either junior, senior or POD team member.
- All users can send and approve each others requests.
- Users can send messages to the connections.
- Users can search for a particular thread.

2. Juniors View:

- Juniors would be able to view threads created by seniors and POD teams.
- Each junior would be uniquely identified by its roll number.
- Junior can attempt assessments posted by seniors or the POD team and view the scores.
- Juniors themselves can read, create, edit or delete their own threads.
- They can either make a post publicly or anonymously.
- Juniors can send approve requests to seniors and POD team members.
- Juniors can search for a particular thread.
- Juniors can view their threads.
- Juniors can message others.

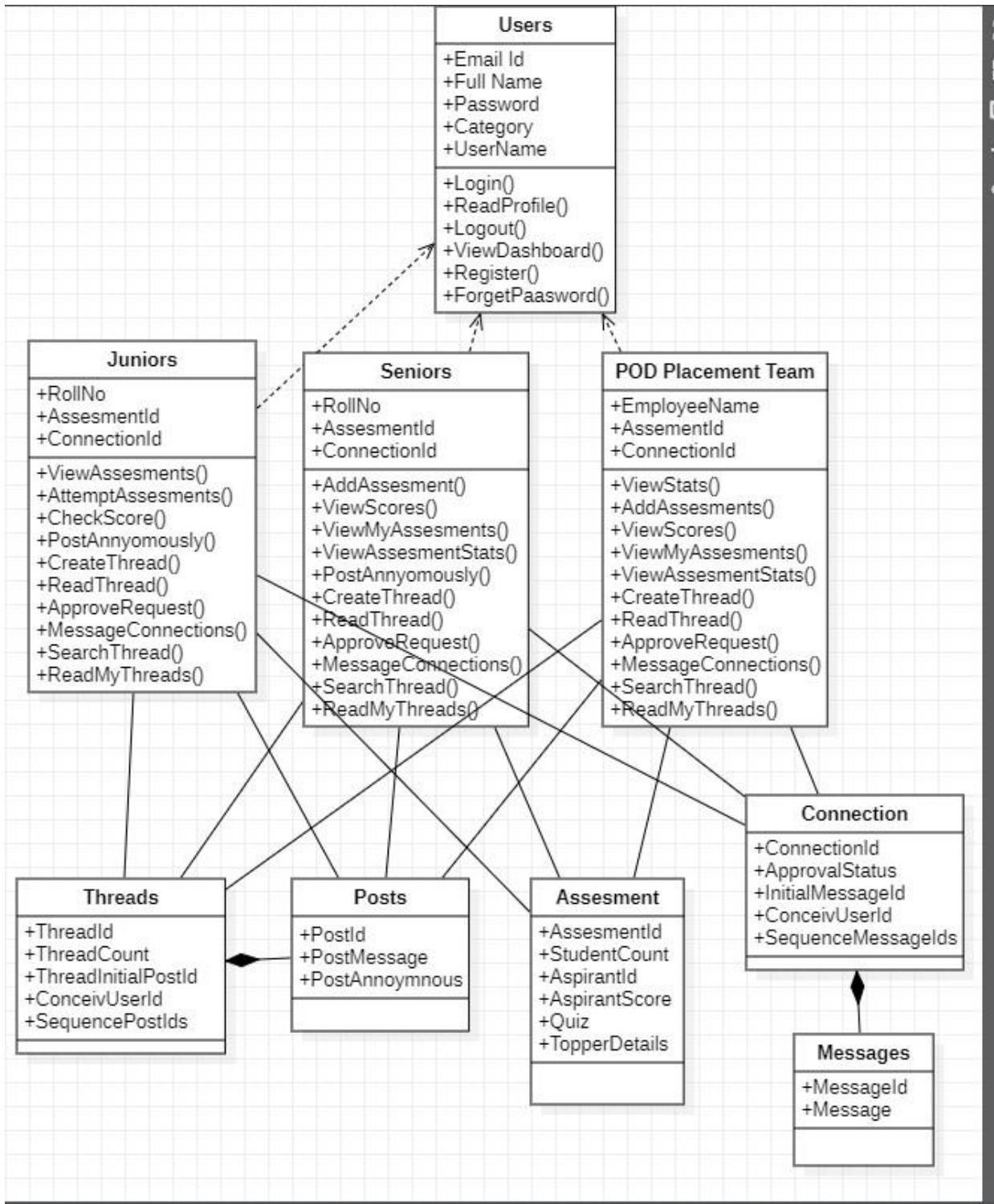
3. Seniors view:

- Each senior would be uniquely identified by its roll number
- Senior can create,edit,delete Threads.
- Seniors can add,edit and delete assessment.
- Seniors can also post publicly or anonymously.
- Seniors can approve requests from Juniors and POD team for connections.
- Seniors can search for threads.
- Seniors can view Assessment Scores and Stats.
- Seniors can view their profile and threads.
- Seniors can message others.

4. POD placement team:

- Team member can view basic stats like number of posts made by a user, assessment attempted by juniors and their scores.
- Team Members can add assessments.
- Team Member can view scores and stats of assessment.
- Team member can create read and update their threads or other users thread.
- Team Member can request and approve connections.
- Team Members can message others.

Class Diagram



Sprint Schedule

Sprint 1: Identify users of the application

Date: 19/06/23 - 25/06/23

Sprint 2: Writing user stories for the requirements

Date: 26/06/23 - 02/07/23

Sprint 3: Creating a storyboard using Storboard That and WireFrame using Figma

Date: 03/07/23 - 10/07/23

Sprint 4: Scheduling project tasks using Jira and creating Gantt Charts, pivot table

Date: 11/07/23 - 18/07/23

Sprint 5: Designing Components and making Class Diagram using StarUML

Date: 19/07/23 - 23/07/23

Sprint 5: Designing Databases and Creating API Endpoints for all Classes

Date: 24/07/23 - 1/08/23

Sprint 6: Designing Test Cases and testing them

Date: 1/08/23 - 03/08/23

Sprint 7: Coding and working on User, Thread Management

Date: 03/08/23 - 10/08/23

Sprint 8: Coding and working on Assessment and Connection Management

Date: 10/08/23 - 17/08/23

Sprint 9: Developing User Interface and fine tuning

Date: 18/08/23-25/08/23

Sprint 10: Making a report on work done from Milestone 1 through Milestone 5

Date: 25/08/23 - 30/08/23

SCRUM Meetings

SCRUM meetings: Every Sunday (10:00 to 12:00) and Wednesday (21:00 to 23:00)

Minutes from Sprint 1 SCRUM meeting:

Discussed about the scope of the project. Identified the primary, secondary and tertiary users of the application. Wrote a few user stories for each type of user , based on the SMART guidelines explained in the lectures.

Minutes from Sprint 2 SCRUM meeting:

Discussed about user stories. Agreed on some of user stories and kept some for further discussion. Deciding about the functionalities of each user based on their role. Went through design heuristics and refined our work. Took each others interview placing ourselves in different user characters.

Minutes from Sprint 3 SCRUM meeting:

This meeting was all about creating a storyboard and wireframe. Here we discussed the flow of the application. Here we finalized the user interface of the application. We focused on making the user interface pleasing as well as intuitive.

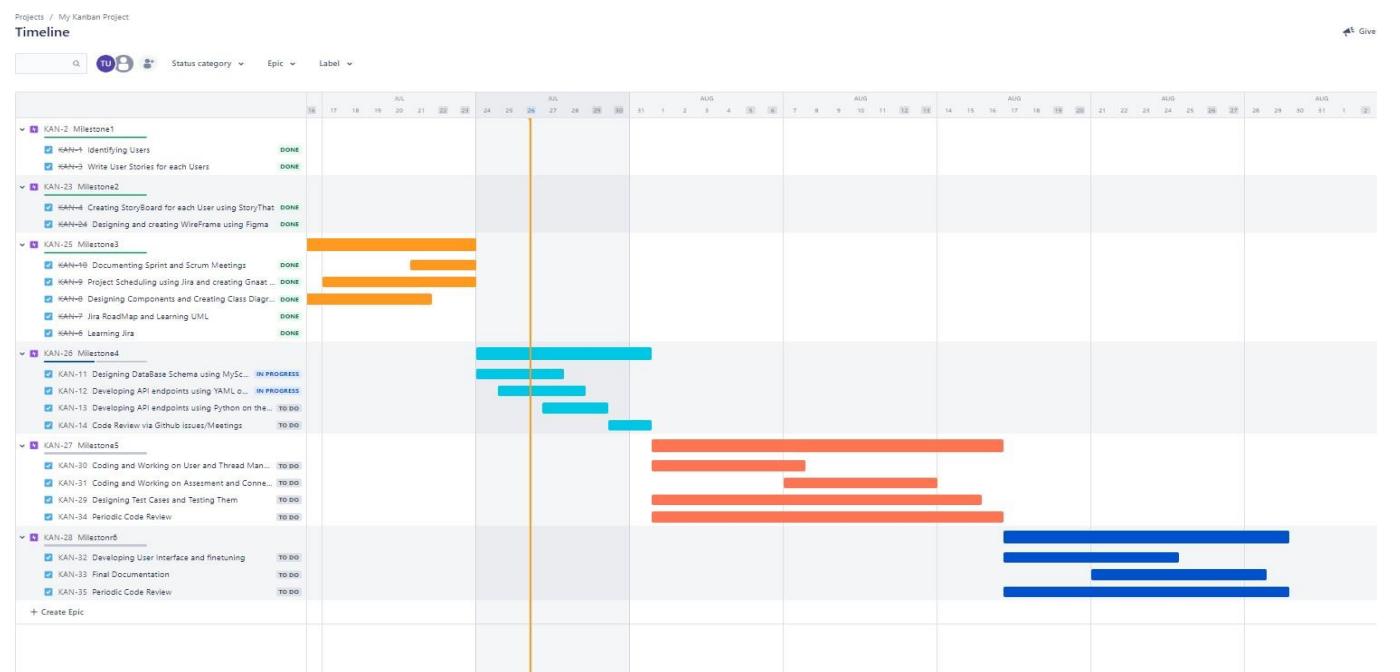
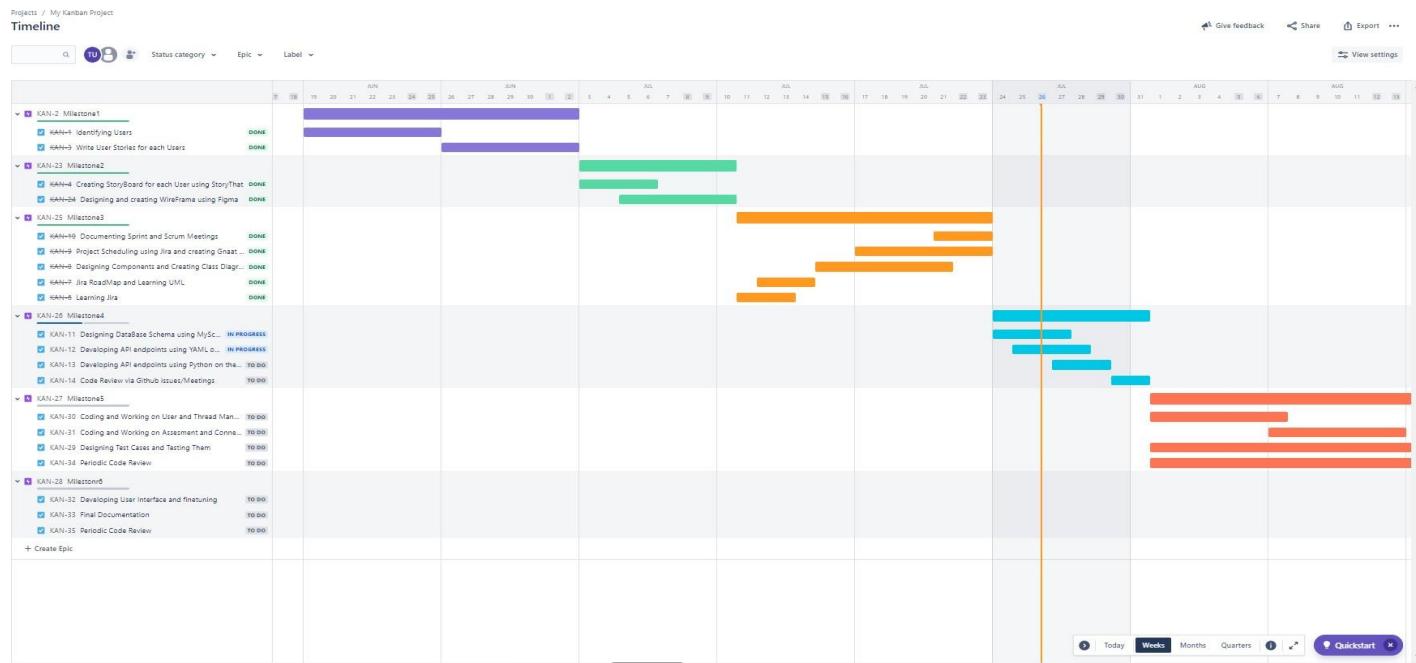
Minutes from Sprint 4 SCRUM meeting:

The team discussed how to use JIRA software to create a Project Schedule pictorially as well as a GANTT chart and the go ahead to start working on it.

Minutes from Sprint 5 SCRUM meeting:

The team discusses about the strategy to approach designing the components and coming up with a Class Diagram using StarUML

Gantt Chart



KanbanBoard

Projects / My Kanban Project

KAN board

TO DO 9		IN PROGRESS 2		DONE 9	
Developing API endpoints using Python on the APP	MILESTONE4	Designing DataBase Schema using MySchema	MILESTONE4	Documenting Sprint and Scrum Meetings	MILESTONE3
<input checked="" type="checkbox"/> KAN-13		<input checked="" type="checkbox"/> KAN-11		<input checked="" type="checkbox"/> KAN-10	
Code Review via Github Issues/Meetings	MILESTONE4	Developing API endpoints using YAML on Swagger	MILESTONE4	Project Scheduling using Jira and creating Gnaat Chart	Milestone3
<input checked="" type="checkbox"/> KAN-14		<input checked="" type="checkbox"/> KAN-12		<input checked="" type="checkbox"/> KAN-9	
Coding and Working on User and Thread Management	MILESTONES	Designing Components and Creating Class Diagram	MILESTONE3	Jira RoadMap and Learning UML	Milestone3
<input checked="" type="checkbox"/> KAN-30		<input checked="" type="checkbox"/> KAN-8		<input checked="" type="checkbox"/> KAN-7	
Coding and Working on Assessment and Connection Management	MILESTONES	Learning Jira	Milestone3	Creating StoryBoard for each User using StoryThat	Milestone2
<input checked="" type="checkbox"/> KAN-31		<input checked="" type="checkbox"/> KAN-6		<input checked="" type="checkbox"/> KAN-4	
Designing Test Cases and Testing Them	MILESTONES	Designing and creating WireFrame using Figma	MILESTONE2	Write User Stories for each Users	MileStone1
<input checked="" type="checkbox"/> KAN-29		<input checked="" type="checkbox"/> KAN-24		<input checked="" type="checkbox"/> KAN-3	
Developing User Interface and finetuning	MILESTONES	Identifying Users	MILESTONE1		
<input checked="" type="checkbox"/> KAN-32		<input checked="" type="checkbox"/> KAN-1			
Final Documentation	MILESTONES				
<input checked="" type="checkbox"/> KAN-33					
Periodic Code Review	MILESTONES				
<input checked="" type="checkbox"/> KAN-34					
Periodic Code Review	MILESTONES				
<input checked="" type="checkbox"/> KAN-35					

Milestone 4

In Milestone 4 we were asked to make API endpoints for the Project on yaml as per the design requirements.

The work done is as below.

Please use this link. https://app.swaggerhub.com/apis/21F1005359_1/placement-portal_api_endpoints/1.0.0 to see the yaml file.

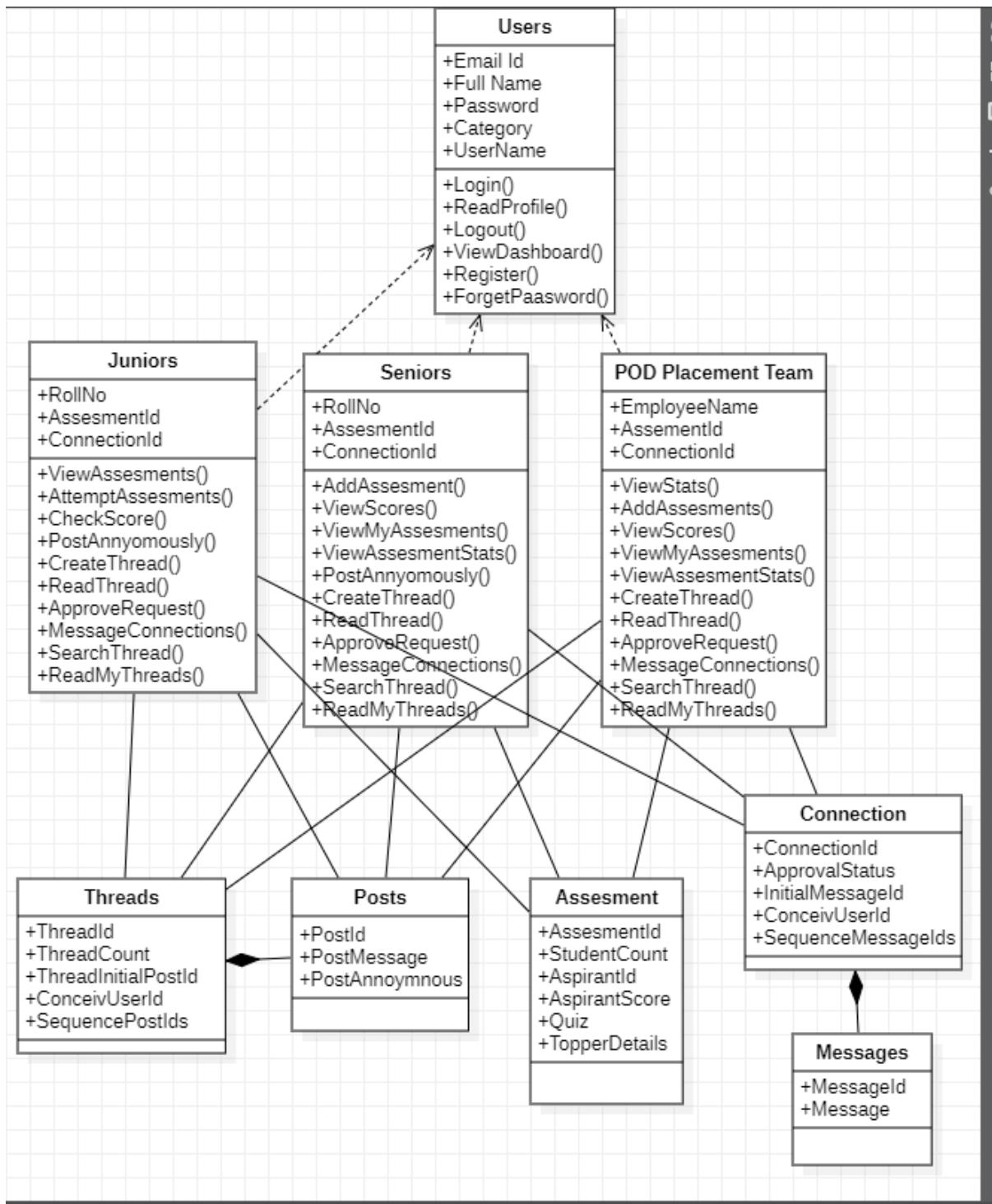
Placement Portal API Endpoints

1.0.0

OAS 3.0

API Documentation for Placement Portal

ER Diagram



Error Codes

Resource	Error Code	Message
Resource	Error Codes	Message
Servers		

user

POST /user

Create user

GET /user/login

Logs user into the system

GET /user/logout

Logs out current logged in user session

GET /user/{user id}

Get user by user name

PUT /user/{user id}

Updated user

DELETE /user/{user id}

Delete user

thread

POST /user/{user id}/thread

Create Thread

GET /user/{user id}/{thread id}

Get Thread Details

PUT /user/{user id}/{thread id}

Update Thread

DELETE /user/{user id}/{thread id}

Delete Thread

post

POST /user/{user id}/{thread id}/Post

Create Post

GET /user/{user id}/{thread id}/{post id}

Get Post Details

PUT /user/{user id}/{thread id}/{post id}

Update Post

DELETE /user/{user id}/{thread id}/{post id}

Delete Post

assesment

POST /user/{user id}/Assesment

Create Assesment

GET /user/{user id}/{assesment id}

Get Assesment Details

PUT /user/{user id}/{assesment id}

Update Assesment

DELETE /user/{user id}/{assesment id}

Delete Assesment

connection

POST	<code>/user/{user id}/Connection</code>
Create Connection	
GET	<code>/user/{user id}/{connection id}</code>
Get Connection Details	
PUT	<code>/user/{user id}/{connection id}</code>
Update Connection	
DELETE	<code>/user/{user id}/{connection id}</code>
Delete Connection	
message	
POST	<code>/user/{user id}/{connection id}/Message</code>
Create Message	
GET	<code>/user/{user id}/{connection id}/{message id}</code>
Get Message Details	
PUT	<code>/user/{user id}/{connection id}/{message id}</code>
Update Message	
DELETE	<code>/user/{user id}/{connection id}/{message id}</code>
Delete Message	

Milestone 5

In this Milestone we were asked to design and describe a few test cases.

The work done is as below.

Group 4

A) A new User account is created at

Page being tested: <http://127.0.0.1:8080/api/user/CreateUser>

Inputs:

Request Method: POST

JSON: {
 "user_id": 26,
 "Name": "ute",
 "email": "w@rw.com",
 "password": "kuk"
 }

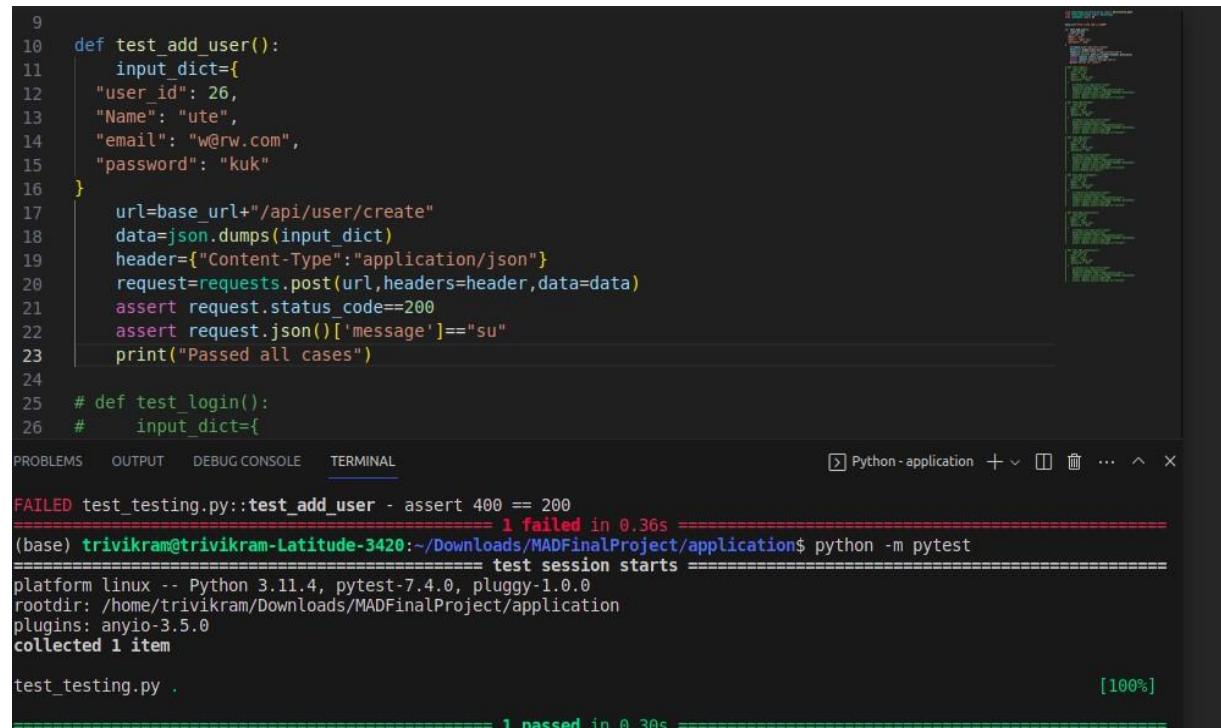
Expected Output:

HTTP Status Code: 200

JSON: { "message": "suc"}

Actual Output:

HTTP Status Code: 200



A screenshot of a terminal window titled "Python - application". The window shows a Python script named "test_testing.py" with code for testing user addition and login. The terminal output indicates a failure in the "test_add_user" test, which asserts a status code of 200 but got 400 instead. The test session starts on a Linux platform with Python 3.11.4, pytest-7.4.0, and pluggy-1.0.0. One item was collected and passed in 0.305 seconds.

```
9
10 def test_add_user():
11     input_dict={
12         "user_id": 26,
13         "Name": "ute",
14         "email": "w@rw.com",
15         "password": "kuk"
16     }
17     url=base_url+"/api/user/create"
18     data=json.dumps(input_dict)
19     header={"Content-Type":"application/json"}
20     request=requests.post(url,headers=header,data=data)
21     assert request.status_code==200
22     assert request.json()['message']=="su"
23     print("Passed all cases")
24
25 # def test_login():
26 #     input_dict={

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
FAILED test_testing.py::test_add_user - assert 400 == 200
=====
(base) trivikram@trivikram-Latitude-3420:~/Downloads/MADFinalProject/application$ python -m pytest
=====
platform linux -- Python 3.11.4, pytest-7.4.0, pluggy-1.0.0
rootdir: /home/trivikram/Downloads/MADFinalProject/application
plugins: anyio-3.5.0
collected 1 item
test_testing.py .
=====
1 passed in 0.305s [100%]
```

B) Login

Page being tested: http://127.0.0.1:8080/api/user_id/password

Inputs:

Request Method: POST

JSON: JSON: { "user_id": 24,
"password": "kuk" }

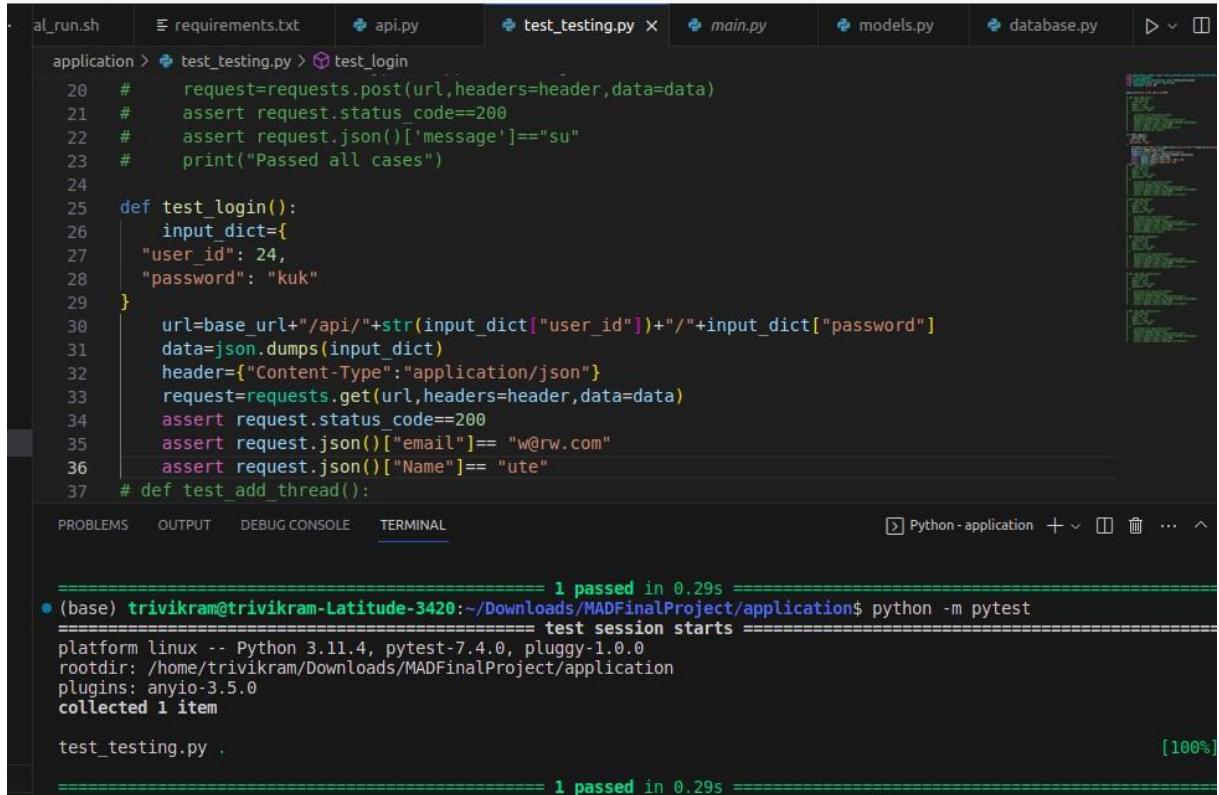
Expected Output:

HTTP Status Code: 200

JSON: { "message": "suc" }

Actual Output:

HTTP Status Code: 200



```

al_run.sh requirements.txt api.py test_testing.py x main.py models.py database.py > v ⓘ
application > test_testing.py > test_login
20     #     request=requests.post(url,headers=header,data=data)
21     #     assert request.status_code==200
22     #     assert request.json()['message']=="su"
23     #     print("Passed all cases")
24
25     def test_login():
26         input_dict={
27             "user_id": 24,
28             "password": "kuk"
29         }
30         url=base_url+"/api/"+str(input_dict["user_id"])+"/"+input_dict["password"]
31         data=json.dumps(input_dict)
32         header={"Content-Type":"application/json"}
33         request=requests.get(url,headers=header,data=data)
34         assert request.status_code==200
35         assert request.json()["email"]=="w@rw.com"
36         assert request.json()["Name"]=="ute"
37     # def test_add_thread():

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Python - application + v ⓘ ... ^
===== 1 passed in 0.29s =====
● (base) trivikram@trivikram-Latitude-3420:~/Downloads/MADFinalProject/application$ python -m pytest
===== test session starts =====
platform linux -- Python 3.11.4, pytest-7.4.0, pluggy-1.0.0
rootdir: /home/trivikram/Downloads/MADFinalProject/application
plugins: anyio-3.5.0
collected 1 item
test_testing.py .
[100%]
===== 1 passed in 0.29s =====

```

C) Create new Thread

Page being tested: http://127.0.0.1:8080/api/user_id/password/CreateThread

Inputs:

Request Method: POST

JSON: { 'user_id':24,
 'thread_id':4,
 'Thread_count':1,
 'Thread_Map':'1:Hi',
 'Title':'Hi'}

Expected Output:

HTTP Status Code: 200

JSON: { "message": "suc"}

Actual Output:

HTTP Status Code: 200

The screenshot shows a code editor with several tabs: api.py, main.py, test_testing.py (active), and models.py. The test_testing.py tab contains the following Python code:

```
application > test_testing.py > test_add_thread
58     # }
59     def test_add_thread():
60         input_dict={
61             'user_id':24,
62             'thread_id':4,
63             'Thread_count':1,
64             'Thread_Map':"1:Hi",
65             'Title':"Hi",
66         }
67         url=base_url+"/"+str(input_dict["user_id"])+"/kuk"+"/CreateThread"
68         print(url)
69         data=json.dumps(input_dict)
70         headers={"Content-Type":"application/json"}
71         request=requests.post(url,headers=headers,data=data)
72         print(request)
73         assert request.status_code==200
74         assert request.json()['message']=="su"
75
```

Below the code editor is a terminal window showing the execution of pytest:

```
(base) trivikram@trivikram-Latitude-3420:~/Downloads/MADFinalProject/application$ python -m pytest> ^C
● (base) trivikram@trivikram-Latitude-3420:~/Downloads/MADFinalProject/application$ python -m pytest
===== test session starts =====
platform linux -- Python 3.11.4, pytest-7.4.0, pluggy-1.0.0
rootdir: /home/trivikram/Downloads/MADFinalProject/application
plugins: anyio-3.5.0
collected 1 item

test_testing.py . [100%]
1 passed in 0.29s
```

D) Create new Post

Page being tested:

http://127.0.0.1:8080/api/user_id/password/thread_idInputs:

Request Method: POST

JSON: { 'thread_id':4,
'Post_Message':'Hello
','Annoynmous':'True',
'Title':'Hi',
'post_id':2}

Expected Output:

HTTP Status Code:

200

JSON: {"message":

"suc"} Actual Output:

HTTP Status Code: 200

Project SUMMARY:

Report on work done in different milestones:

- Milestone 1 :The group identified the different users of the application in terms of the primary, secondary and tertiary users. Post the same, user stories (following SMART guidelines) were written for the different users identified earlier. 25 user stories which had the mandatory features as suggested by the project and some additional features as deemed necessary were compiled and submitted before the deadline.

- Milestone 2 : Four storyboards (using Comicgen and Google Slides) were depicted using comic strips:

- Many storyboards were made from the perspective of a junior.
 - Many storyboard was made from the perspective of a senior.
 - Many storyboard was made from the perspective of a pod/admin team.

Wireframes were designed for the application using the design heuristics that were learnt in the Software Engineering course. Essential usability principles that were incorporated in our wireframe was as follows:

- Effectiveness: To ensure that users can carry out their work efficiently.
 - Efficiency: Each user can effectively carry out their task in a minimal number of steps.
 - Safety: Confirmation prompts to prevent any unwanted actions
 - Learnability: Useful tool-tips on mouse hover
 - Memorability: Easy to use, consistent interface, which helps the user to remember the same.

Some design heuristics that were followed during this phase were:

- Consistency: Layout of different pages are similar.
 - Clean and Functional Design: Clutter free easy to recognize design which focuses on features and functionality, with pleasant aesthetics.

Provide Help: Useful tool-tips on mouse hover.

- Milestone 3 : A rough project schedule was drawn upon keeping in mind the deadlines for each milestone and the amount of work that was required for the same. Sprints were planned for the different tasks.Jira was chosen as an appropriate project management tool and a Gantt Chart roadmap was plotted to keep track of the progress.Different components were designed as UML diagrams using the Star UML software to model the software system. Details of scrum meetings conducted up until this stage were presented as “minutes of the meet” in the milestone 3 submission.Project Scheduling and estimates were released regarding how and by when the Project should progress.

- Milestone 4: In this week we learnt about the designing of api endpoints how to connect it to the python backend.How to use Swagger to publish your api on

the web, how to use Mermaid to make an ER diagram. Post designing the components and going through the entire idea. API endpoints were made using Insomnia, Swagger, Mermaid Ink. Database were also made on DbBrowser MySql. The components were translated to databases. As per the schedule the Project progresses. Many iterations were exhausted to nail the perfect API endpoint and database schema. API's for Users, Threads, Posts, Assessments, Aspirants, Connections and Messages were made and released.

- Milestone 5: In this Milestone we learnt about testing and using test cases for the project. We learnt about pytest and what are the different types of testings that are present. Various test cases were considered to check the performance of the system in the context to the API without implementing in real life as was the feature of pytest. Various test cases were implemented to check login functionality, dashboard functionality, assessment functionality and thread and post functionality i.e get of each to check if the data was retrieved correctly.
- Milestone 6: This was the final milestone and the Project submission. In the weeks assigned to these we learnt about practices that are used in coding an application, how to communicate effectively how to deploy the application using Heroku how to use issues trackers and pull requests for the same. The entire codebase was developed right from backend to frontend. The integration took time but with effective use of Pytest API endpoints were tested thoroughly before. The codebase was uploaded on github. The entire work was completed as per the Project Estimate. Issue trackers and pull requests were also created on github for the project as it was built. The entire application was deployed. Documentation such as these were carried out during this Milestone. The Project concluded with ticking all the estimates as before. Vuejs was used for frontend while flask for backend.

Implementation Details of the Project

- Technologies and tools used
 - Technologies for the backend
 - Flask
 - Flask Restful (For creating API endpoints)
 -
 - Technologies for the frontend
 - Vue 3 CLI
 - JavaScript
 - Vue Router
 -

- General technologies used
 - GitHub (for versioning, code management, tracking, reviewing, issues, etc.)
 - Insomnia, Swagger (for yaml coding of the apis)
 - Visual Studio editor(for writing the entire code)
 - Chatgpt (for optimizing the code)
 - Comic gen(for creating user board)
 - Pytest (for testing the apis)
 - Jira (for project management)
-

- **Hosting:** info regarding hosting

- Instructions to run the application

- On Ubuntu/MAC OS:
 - Git clone the repository.
 - Change the directory to the “backend” directory inside the “Milestone-6-Final-Submission” directory using the command :
cd ./Milestone-6-Final-Submission/Code/backend
 - Just enter the command “sh local-setup.sh” for setting up the environment for installing all the required libraries and so forth.
 - Enter the command “sh local-run.sh” for running the backend server.
 - Open new terminal direct to the “frontend” folder go inside the “client” folder ,write the command “npm install” to create all the nodepackages and dependencies associated with it.
 - Run the command “npm run serve” to run the frontend server and open the url “localhost//8080” .

- On Windows:
 - Git clone the repository.
 - Change the directory to the “backend” directory inside the “Milestone-6-Final-Submission” directory using the command :
cd .\Milestone-6-Final-Submission\Code\backend.

- Just enter the command “sh local-setup.sh” for setting up the environment for installing all the required libraries and so forth.
- Enter the command “sh local-run.sh” for running the backend server.
- Open new terminal direct to the “frontend” folder go

inside the “client” folder ,write the command “npm install” to create all the nodepackages and dependencies associated with it.

- Run the command “npm run serve” to run the frontend server and open the url “localhost//8080”.

Code Review ,Issue Reporting and Tracking

This was done on github.

ISSUES:

The screenshot shows a GitHub Issues page for the repository "TrivikramUmanath/Placement-Portal-FINAL". The URL in the address bar is "github.com/TrivikramUmanath/Placement-Portal-FINAL/issues?q=is%3Aissue+is%3Aclosed". The search bar contains the query "is:issue is:closed". There are 9 labels and 0 milestones. A "New issue" button is visible. The page lists 7 closed issues:

- Parse Int not working for assesment id (#7)
- UML discrepancies (#6)
- Unable to tag milestones in Jira (#5)
- Error 404 for Connection API (#4)
- Is Aspirant table required? (#3)
- Issue with the Apis (#2)
- Login page functionality isn't right (#1)

Each issue has a checkbox, a title, a description, and a comment count (e.g., 2, 1, 1, 2, 2, 4, 2).

PULL REQUEST:

github.com/TrivikramUmanath/Placement-Portal-FINAL/pulls

TrivikramUmanath / Placement-Portal-FINAL

Code Issues Pull requests 4 Actions Projects Security Insights Settings

Filters Q is:pr is:open Labels 9 Milestones 0 New pull request

Author	Label	Projects	Milestones	Reviews	Assignee	Sort
TrivikramUmanath						
TrivikramUmanath						
TrivikramUmanath						
TrivikramUmanath						

4 Open 0 Closed

Add files via upload #11 opened now by TrivikramUmanath

Add files via upload #10 opened 4 minutes ago by TrivikramUmanath

Add files via upload #9 opened 5 minutes ago by TrivikramUmanath

Delete Placement_Portal.mp4 #8 opened 10 minutes ago by TrivikramUmanath

ProTip! no:milestone will show everything without a milestone.

Presentation of the Working Prototype

Github repository:

<https://github.com/TrivikramUmanath/Placement-Portal-FINAL>

Presentation video:

https://drive.google.com/file/d/1XOwNVvkYfV3CPAedAkOA TNfjTni_SLTj/view?usp=sharing

Deployment Link:

