SOFTWARE ENGINEERING PROJECT

Quick Resolve

GROUP 11

Jemma Mariya George
R. No.: 21f1001937

Barun Kumar Sinha
R. No.: 21f1002021

Vidushi Talwar
R. No.: 21f1004809

Acknowledgement

We would like to express our sincere gratitude to our mentors and guides, Dr Sridhar Iyer and Dr Prajish Prasad, for giving us an opportunity to work under their guidance on this project. We would also like to thank the course team, Mr. Arup Chattopadhyay and Mr. Ankur Parmar for their constant support and guidance.

We truly appreciate and value their esteemed guidance and encouragement throughout the project. We are highly indebted for their precious advice, perceptive guidance and constant words of encouragement.

Finally, we would also like to thank IIT Madras for giving us an opportunity and platform to convert this project into a reality.

Jemma Mariya George **Barun Kumar Sinha** **Vidushi Talwar**

Certificate

This is to certify that Group 11 - Jemma Mariya George, Barun Kumar Sinha & Vidushi Talwar – students of B.S. in Data Science and Applications, have submitted the project entitled 'Quick Resolve' for the partial fulfilment of the requirements of Software Engineering project.

It embodies the work done by them during the Software Engineering course under the due supervision of Dr Sridhar Iyer and Dr Prajish Prasad.

Dr Sridhar Iyer **Dr Prajish Prasad**

Index

S.No.	Topic	Page No.
	Acknowledgement	1
	Certificate	2
1	Introduction	
	Overview	4
	Objective	5
	Problem Statement	6
2	Project Management	
	Process Model	7
	Risk Management	8
	Risk Assessment	9
	Project Schedule	10
	Minutes of Scrum Meetings	13
3	Requirement Gathering & Analysis	
	Identifying Users	21
	User Stories	24
4	Design	
	Storyboards	28
	Paper Prototypes	47
	Component Design	57
	Class Diagrams	64
5	Development	
	API Endpoint Description	65
	Test Driven Development	88
	Issue Tracking & Code review	103
6	Deployment	106
7	Technologies & Tools used	107

Overview

Quick Resolve is a flask based software that provides a platform to facilitate quick & effective query addressal.

Quick Resolve can be used by the Support team to prioritize the important / common queries first, while ensuring no duplicate queries are created.

In order to build software Agile Methodology has been adopted. The front-end of the software has been developed using Python language and the back end has been designed using Flask SQL-Alchemy.

This software provides the following facilities:

- User Login
- User Registration
- Ticket Creation
- Ticket Resolution
- View Similar Tickets
- +1 Existing Tickets
- Tickets sorted acc. to priority
- Pin Tickets
- Delete Tickets
- Add Ticket to FAQ
- Ticket resolution Notifications
- Making Announcements

Objective

The main objective of this application is to assist the IITM BSc degree Support Team.

Quick Resolve aims to:

- Help students create tickets with ease
- Help support team address & resolve tickets
- Notify students once their queries have been resolved
- Show existing tickets created in categories of interest to students
- Help students like or +1 existing tickets instead of creating duplicate tickets again and again
- Help support team view the unresolved tickets sorted according to their priority so that popular queries are addressed first
- Enable support team to pin important tickets so that students can go through the same
- Help support team make announcements
- Enable admin to delete tickets violating the code of conduct
- Enable users to report tickets deemed inappropriate
- Help add important & relevant tickets to FAQ document

Problem Statement

The support team at the IITM BSc degree program often get overwhelmed with emails from students regarding queries and concerns. Your task is to create an online support ticketing system for the IITM BSc degree program. Students can create a support ticket for a particular concern or query. Before they create a ticket, the system should also show a list of similar tickets, and allow users to like or +1 an already existing support ticket, so that duplicates are not created. This way popular concerns or queries can be prioritised by the support team.

After the support team addresses the concern, they can mark the ticket as resolved, and an appropriate notification should be sent to concerned users.

Another important feature of the ticketing system is dynamic FAQ updation. Many student concerns can be FAQs which will be useful for future students. If appropriate, the support query and response should be added to the FAQ section by support admins, and appropriately categorised, so that an updated FAQ will be readily available to students. The platform should allow users to enroll as students, support staff and admins.

Process Model

The software model that we have used to develop our application-”Quick Resolve” software is the Agile Model.

The Agile Methodology is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches.

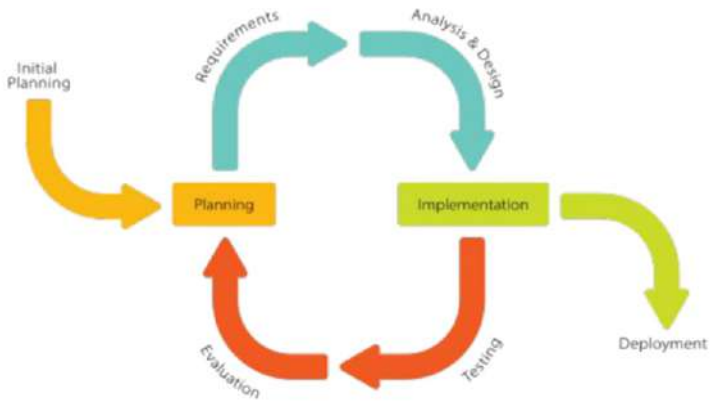
Rather than just following approaches and practices, the Agile Philosophy focuses more on adhering to the broad philosophy.

The Agile Manifesto specifies 4 key principles:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

While the traditional "waterfall" approach has one discipline contribute to the project, then "throw it over the wall" to the next contributor, Agile calls for collaborative cross-functional teams. Open communication, collaboration, adaptation, and trust amongst team members are at the heart of agile.

Agile isn't defined by a set of ceremonies or specific development techniques. Rather, agile is a group of methodologies that demonstrate a commitment to tight feedback cycles and continuous improvement.



Risk Management

A risk is an anticipated unfavourable event or circumstance that can occur while a project is underway. Risk Management is an essential component of Project Management due to the Intangible Nature of software.

The potential risks in software development can be categorized as:

1. Technical Risks
   – Related to the technical aspects of the project
   – Can occur due to the development team's insufficient knowledge about the product.
2. Project Risks
   – Occur due to problems in budget, schedule, personnel, resources, and customer-related issues
3. Business Risks
   – Hold the potential of harming the business aspects of the software product.

Events such as Conflicts in a team can hamper the software development. Thus it is imperative to assess all possible outcomes and try to take measures to mitigate the risk.

In order to assess the risks, the project manager creates what's known as a "Risk Table". Within a risk table, the project manager assigns a probability of occurrence to the estimated risks alongside its estimated impact measured on a scale of 1- 4, 1 being negligible and 4 being catastrophic.

With this estimated information, the Risks are then computed using the formula:

Risk = Probability x Impact

Risk Assessment

S.No.	Risk	Category	Probability	Impact	Risk
1	Lack of Experience / Insufficient knowledge	Project Risk	70%	4	2.8
2	Schedule Slippage	Project Risk	60%	4	2.4
3	Developing wrong functions & UIs	Technical Risk	50%	4	2
4	Personnel Shortfall	Project Risk	50%	3	1.5
5	Shortcomings in external components	Technical Risk	40%	3	1.2
6	Conflict in team members	Project Risk	20%	3	0.6
7	Product not competitive in market	Business Risk	10%	4	0.4

Project Schedule

Project Scheduling is as important as the project development. Creating a schedule for a project not only helps in monitoring the timely completion of a task but also in taking corrective action on time.

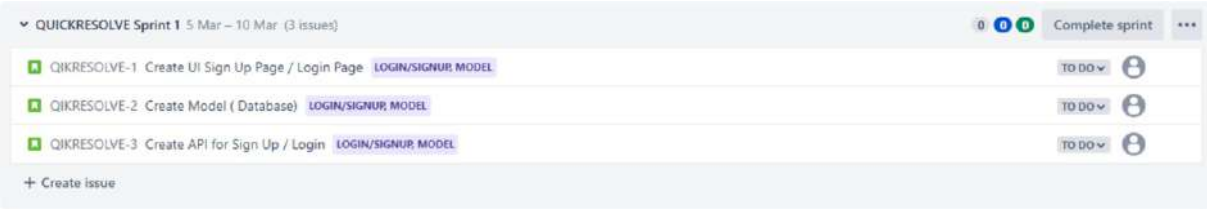
The main activities in scheduling include:

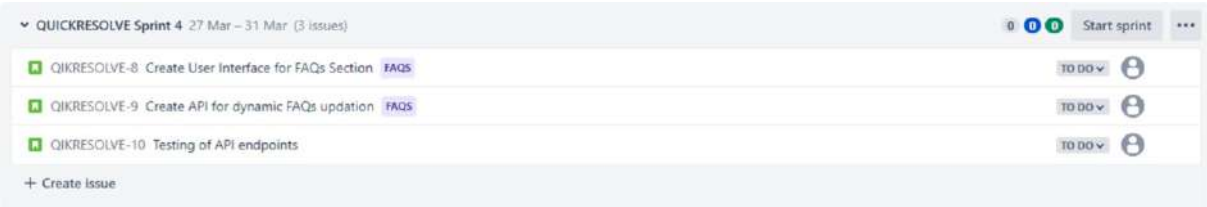
- Identifying the major activities
- Breaking down each activity into tasks
- Determining the dependency among different tasks
- Estimating the time durations required for completing the tasks
- Representing the information using charts, graphs, networks etc.
- Determining the start and end dates
- Determining the critical path
- Allocating resources to the tasks

There are a variety of tools which can help in Project Scheduling. For the purpose of the project, we have used **JIRA** to prepare the Project Schedule.

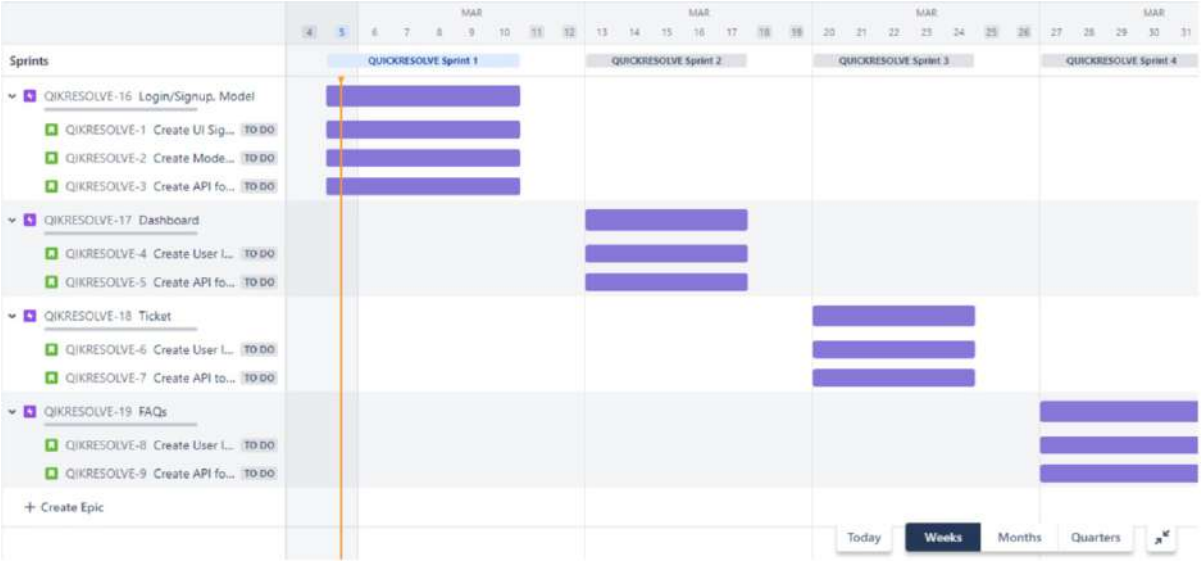
JIRA is a bug tracking tool that allows software developers to plan, track and work faster. Being an agile project management tool, it supports any agile methodology, be it scrum, kanban or any unique flavour. Agile Boards, Backlogs, roadmaps, reports assist in planning, tracking & managing agile projects, all using 1 tool!

Project Schedule:
Quick Resolve - Sprints





Project Schedule:
Quick Resolve - Gantt Chart



Scrum Meetings

Also known as Daily scrum or 'Standup', a Scrum meeting serves the purpose of sharing an individual's progress & estimating the level of completeness until the next scrum meeting.

Scrum meetings require everyone involved in the product's development to attend. This includes the development team, scrum master & the product owner.

Herein each member asks themselves **3 questions:**

- What did I work on yesterday?
- What am I working on today?
- What issues are blocking me?

Benefits of scrum meetings:

- Enables the team to be in sync on how things are going
- Allows for identification and solution of impediments
- Provides an opportunity for small course corrections within the sprint
- Building trust between team members
- High visibility of progress
- Promotes self-organization in team: team members hold each other accountable for achieving their daily commitments.

Minutes of the Meetings: Iteration-1

Location:	**WhatsApp Group**
Date:	19th February 2023 (Sprint Iteration 1)
Time:	10:00 am – 10:15 am
Attendees:	Jemma Mariya George, Barun Kumar Sinha, Vidushi Talwar
Project Start:	18th February 2023

Agenda:

- Review tasks accomplished yesterday
- Plans for the day
- Roadblocks hampering progress

Team Member	Tasks Accomplished (on 18th Feb, 2023)	Status	Problems Encountered	Plans for the Day
Jemma Mariya George	Finished User Requirements Document	**Complete**	–	Submit Milestone 1, Honour Code
Barun Kumar Sinha	Finished User Requirements Document	**Complete**	–	Submit Honour Code
Vidushi Talwar	Finished User Requirements Document	**Complete**	–	Submit Honour Code

Key Notes: It was decided that the milestones will be committed roll number wise, such that each member would have committed 2 milestones by the end of the project (6 weeks).

Minutes of the Meetings: Iteration-2

Location:	**WhatsApp Group**
Date:	20th February 2023
Time:	11:45 am – 12:00 pm
Attendees:	Jemma Mariya George, Barun Kumar Sinha, Vidushi Talwar
Project Start:	18th February 2023

Agenda:

- Review tasks accomplished yesterday
- Plans for the day
- Roadblocks hampering progress

Team Member	Tasks Accomplished (on 19th Feb, 2023)	Status	Problems Encountered	Plans for the Day
Jemma Mariya George	Submit Milestone 1	**Complete**	–	Paper Prototype
Barun Kumar Sinha	Submit Honour Code	**Complete**	–	Milestone 2 Document
Vidushi Talwar	Submit Honour Code	**Complete**	–	Storyboard

Key Notes: The work for the 2nd milestone was split among the members & the 1st milestone was committed along with the honour codes on the classroom.

Minutes of the Meetings: Iteration-2

Location:	**WhatsApp Group**
Date:	21st February 2023
Time:	10:00 am – 10:15 am
Attendees:	Jemma Mariya George, Barun Kumar Sinha, Vidushi Talwar
Project Start:	18th February 2023

Agenda:

- Review tasks accomplished yesterday
- Plans for the day
- Roadblocks hampering progress

Team Member	Tasks Accomplished (on 20th Feb, 2023)	Status	Problems Encountered	Plans for the Day
Jemma Mariya George	Paper Prototype	**In Progress**	–	Complete Paper Prototype
Barun Kumar Sinha	Milestone 2 Document	**Complete**	–	Start working on Milestone 3
Vidushi Talwar	Storyboard	**Complete**	–	Start working on Milestone 3

Key Notes: Each member shared their respective progress and discussed the problems encountered if any.

Minutes of the Meetings: Iteration-2

Location:	**WhatsApp Group**
Date:	26th February 2023
Time:	10:00 am – 10:15 am
Attendees:	Jemma Mariya George, Barun Kumar Sinha, Vidushi Talwar
Project Start:	18th February 2023

Agenda:

- Review tasks accomplished yesterday
- Plans for the day
- Roadblocks hampering progress

Team Member	Tasks Accomplished (on 25th Feb, 2023)	Status	Problems Encountered	Plans for the Day
Jemma Mariya George	Paper Prototype	**Complete**	–	Start Milestone 3
Barun Kumar Sinha	Milestone 3	**In progress**	–	Submit Milestone 2
Vidushi Talwar	Milestone 3	**In progress**	–	Milestone 3

Key Notes: Each member shared their respective progress and discussed the problems encountered if any.

Minutes of the Meetings: Iteration-3

Location:	**WhatsApp Group**
Date:	27th February 2023
Time:	12:00 pm – 12:15 pm
Attendees:	Jemma Mariya George, Barun Kumar Sinha, Vidushi Talwar
Project Start:	18th February 2023

Agenda:

- Review tasks accomplished yesterday
- Plans for the day
- Roadblocks hampering progress

Team Member	Tasks Accomplished (on 26th Feb, 2023)	Status	Problems Encountered	Plans for the Day
Jemma Mariya George	Start Milestone 3	**In progress**	–	Designing Components
Barun Kumar Sinha	Submit Milestone 2	**Complete**	–	Project Schedule, Class Diagrams
Vidushi Talwar	Start Milestone 3	**In progress**	–	Minutes of the Meetings

Key Notes: The work for the 3rd milestone was split among the members & the 2nd milestone was committed on the classroom.

Minutes of the Meetings: Iteration-3

Location:	**WhatsApp Group**
Date:	28th February 2023
Time:	11:30 am – 11:45 am
Attendees:	Jemma Mariya George, Barun Kumar Sinha, Vidushi Talwar
Project Start:	18th February 2023

Agenda:

- Review tasks accomplished yesterday
- Plans for the day
- Roadblocks hampering progress

Team Member	Tasks Accomplished (on 27th Feb, 2023)	Status	Problems Encountered	Plans for the Day
Jemma Mariya George	Designing Components	**In progress**	–	Designing Components
Barun Kumar Sinha	Project Schedule, Class Diagrams	**In progress**	–	Project Schedule, Class Diagrams
Vidushi Talwar	Minutes of the Meetings	**In progress**	–	Minutes of the Meetings

Key Notes: Each member shared their respective progress and discussed the problems encountered if any.

Minutes of the Meetings: Iteration-3

Location:	**WhatsApp Group**
Date:	05th March 2023
Time:	11:00 am – 11:15 am
Attendees:	Jemma Mariya George, Barun Kumar Sinha, Vidushi Talwar
Project Start:	18th February 2023

Agenda:

- Review tasks accomplished yesterday
- Plans for the day
- Roadblocks hampering progress

Team Member	Tasks Accomplished (on 04th March, 2023)	Status	Problems Encountered	Plans for the Day
Jemma Mariya George	Designing Components	**Complete**	–	Start working on milestone 4
Barun Kumar Sinha	Project Schedule, Class Diagrams	**Complete**	–	Start working on milestone 4
Vidushi Talwar	Minutes of the Meetings	**Complete**	–	Submit Milestone 3

Key Notes: The work for the 4th milestone was split among the members & the 3rd milestone was committed on the classroom.

Identifying Users

The first step towards the development of any software is Requirement Gathering and Analysis.

But before we can start gathering the user requirements, it is important to identify the users of our system.

The users can be broadly classified into 3 categories:

1. **Primary Users:**
   The frequent users of the system. For example, independent sellers in the case of Amazon Seller Portal.

2. **Secondary Users:**
   These would include those who don't use the system directly but rather through an intermediary. For example, the secondary users of the Amazon Seller Portal would include the sales managers.

3. **Tertiary Users:**
   The Tertiary Users consists of those who do not use the software at all. But they are affected by the introduction of the software or may even influence the purchase of the software. For example, when talking about the seller portal, the tertiary users would include users such as the Banks & Buyers.

Users: Ticketing System

The Potential Users for the Ticketing System include:

- **Primary Users**:
  - **Students** – The Ticketing System provides a platform wherein the students can get their queries resolved, hence they can be classified as a frequent user.
  - **Support Staffs** – Herein, the support staff is responsible for addressing the concerns of the students & resolve them to the best of their abilities. Thus, they can be classified as frequent users as well.
  - **Admins** – The admin is responsible for ensuring that the interactions strictly adhere to the community guidelines. They need to keep a close eye at the platform, and hence can be categorized as primary users.

- **Secondary Users**:
  - **Potential Students** – The Ticketing System provides a feature using which the FAQs are updated. The potential students looking for information on the institute are likely to refer to the FAQs, thereby indirectly using the platform.
  - **BSc Degree Coordinators** – The BSc Degree Coordinators want to have periodic reports on the concerns raised by the students. These reports can be created efficiently using the Ticketing System since it is a platform wherein the students raise their concerns. Hence they can be categorized as secondary users of the system.
  - **Course Faculty** – The Course Faculty would want regular updates to understand the queries and concerns of the students regarding the Course content so as to take appropriate measures (if any required).

- **Tertiary Users:**
  - **Search Engines** – Though the search engines do not use the system directly or indirectly, they influence the use of the software. Based on the results displayed by the search engines, the users will be able to access the application. Hence making them eligible to be referred to as tertiary users.
  - **Parents** – Though parents would not use the system in any way, they influence the use of the software by encouraging & inspiring their children to take up the course and learn more about it.

- **Data Analysts and R&D team** – They influence the use of the software through the kind of inferences they draw regarding the course.
- **Training & Placement Cell** – They tend to influence the use of the software as students joining the program would like to understand the training and placement records of the degree. Higher the number of placements, more are the chances of people being drawn towards joining the program, hence driving them to use the software.
- **Third Party Apps** – The Third Party Apps such as Shiksha.com, etc. do not use the system in any way but influence the use of the software by providing reviews about the institute & driving the students to explore the university, the courses offered & thus the FAQ document.
- **IT support staff for deployment & maintenance** – The IT support staff does not use the system at all but influences the use of the software. From deployment to maintenance, the support staff ensures that the system is durable & up to the user standards. Thus, they can be classified as tertiary users as well.

User Stories

The agile life cycle begins with the Behavior Driven Design. Herein we ask questions about the behavior of an application before and during the development. The Requirements are continuously refined to meet our expectations. The BDD Version of requirements is User Stories.

The User Stories take the place of what the SRS was in the plan and document perspective. These are short, informal and plain language description of what a user wants to do within a software product which is of value to them.

A user story is the smallest unit of work which can be done in 1 sprint. It should follow the 'Role-feature-benefit' pattern:

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

Benefits:

- Lightweight
- Help plan and prioritize development
- Concentrate on behavior vs implementation of the application
- Conversation between users and the development team.

As per the **SMART Guidelines**, the User Stories should be:

- **Specific** – They shouldn't be vague & at 1 glance we should be able to understand what exactly needs to be implemented.
- **Measurable** – Each story should be testable, which implies that there are known expected results for some inputs.
- **Achievable** – Ideally, we should be able to implement the user story in 1 agile iteration (1-2 weeks). If this is not possible, we should subdivide the stories into smaller ones.
- **Relevant** – The user stories should be relevant, i.e., they should have some business value to one or more stakeholders. This can be achieved by asking questions like, 'why', 'so what'.
- **Timeboxed** – Stop implementing a feature once the time budget exceeds. In such a situation we have 3 options:
    - Give up
    - Divide the story into smaller ones
    - Reschedule the leftover work

User Stories: Ticketing System

The **Primary User Stories** for the Ticketing System include:

Primary User Stories- Students:

- As a **Student**,
  I want to **enroll & login to the system using username and password**,
  So that I can **get access to the facilities** such as putting forth my queries & concerns.
- As a **Student**,
  I want to **search for existing tickets by specifying the topics or keywords**,
  So that **Duplicate Tickets aren't created.**
- As a **Student**,
  I want to be able to **like or +1 the existing tickets**,
  So that **Duplicate Tickets aren't created and at the same time the support staff can identify those queries as popular or of high priority.**
- As a **Student**,
  I want to **create a ticket** for my query / concern **under the relevant topics**,
  So that the **support team can address my concerns.**
- As a **Student**,
  I want to be **notified within 1 minute when my concerns are addressed**,
  So that I can get my **queries resolved as soon as possible.**
- As a **Student**,
  I want to **see the queries posted by my fellow learners under the relevant topics & sub-topics**,
  So that I can **go through them & learn from their posts.**
- As a **Student**,
  I want to see the **important discussions pinned to the top of my screen**,
  So that I **don't miss out on the important information.**

Primary User Stories- Support Staff:

- As a **Support staff**,
  I want to **login to the system using the username and password**,
  So that I can **get access to the facilities** such as addressing user queries.
- As a **Support staff**,
  I want to be **identified as a support team member**,
  So that **students can trust my responses as verified & valid.**
- As a **Support staff**,
  I want to get a **list of the tickets arranged as per their popularity**,
  So that **high priority concerns are addressed first.**

- As a **Support staff**,
  I want that the **tickets shouldn't be duplicated**,
  So that I **don't waste time** answering the same query again & again.
- As a **Support staff**,
  I want to **mark the tickets as resolved** once I've addressed them,
  So that **students can be informed regarding the same.**
- As a **Support staff**,
  I want the **popular queries to be added to the FAQs**,
  so that it can **help address the concerns of the future students thereby simplifying my work.**
- As a **Support staff**,
  I want to be able to **pin the important queries once I've resolved them**,
  So that **students don't miss the important information.**

Primary User Stories- Admins:

- As an **Admin**,
  I want to **login to the system**,
  so that I can **access the facilities & get the necessary controls.**
- As an **Admin**,
  I want to be able to **add the appropriate resolved tickets to the FAQs**,
  So that the **FAQs prove to be more useful** for future students.
- As an **Admin**,
  I want **controls such as reviewing, deleting, reporting the tickets**,
  So that it can be **ensured that the conversations taking place adhere to the community guidelines.**

The **Secondary User Stories** for the Ticketing System include:

- As a **Potential Student**,
  I want an **FAQ document**,
  So that I can **get all the information I need and make an informed decision.**
- As a **BSc Degree Coordinator**,
  I want a **periodic report on the concerns raised by the students**,
  So that I can **look into them & take appropriate measures**, if any required.

The **Tertiary User Stories** for the Ticketing System include:

- As a **Search engine**,
  I want to be able to **display the appropriate resources**,
  So that **people can find what they are looking for** & get their queries resolved.

- As a **Third party app**,
  I want to be able to **show all required information about institutions**,
  So that **people can compare different universities & make informed decisions.**
- As an **IT support staff**,
  I want to **deploy and maintain the software**,
  So that the **various stakeholders can access, use it with ease.**

Prototyping

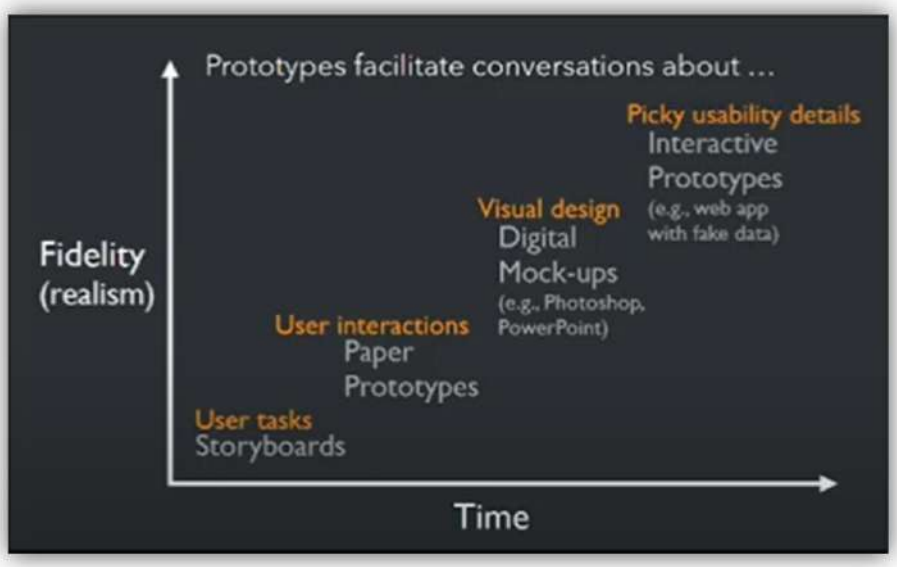
Since the initial design ideas might not be the best ones, & developing an interface takes time, money & effort; thus we start the UI design phase by developing 'Prototypes'.

Prototypes allow us to quickly test on users, get feedback, iterate and pivot. They answer questions and support designers in choosing between the alternatives.

Purpose:

- To test out the technical feasibility of an idea
- Clarify some vague requirements
- User testing and evaluation

Types of Prototypes include:



Storyboard

A ‘Storyboard’ is a **hand-drawn comic** that features:

Setting + Sequence + Satisfaction

The “Setting” component includes:

- People Involved
- Environment
- Task being accomplished

The “Sequence” component answers questions such as:

- What steps are involved?
- What leads someone to use the app?
- What task is being illustrated?

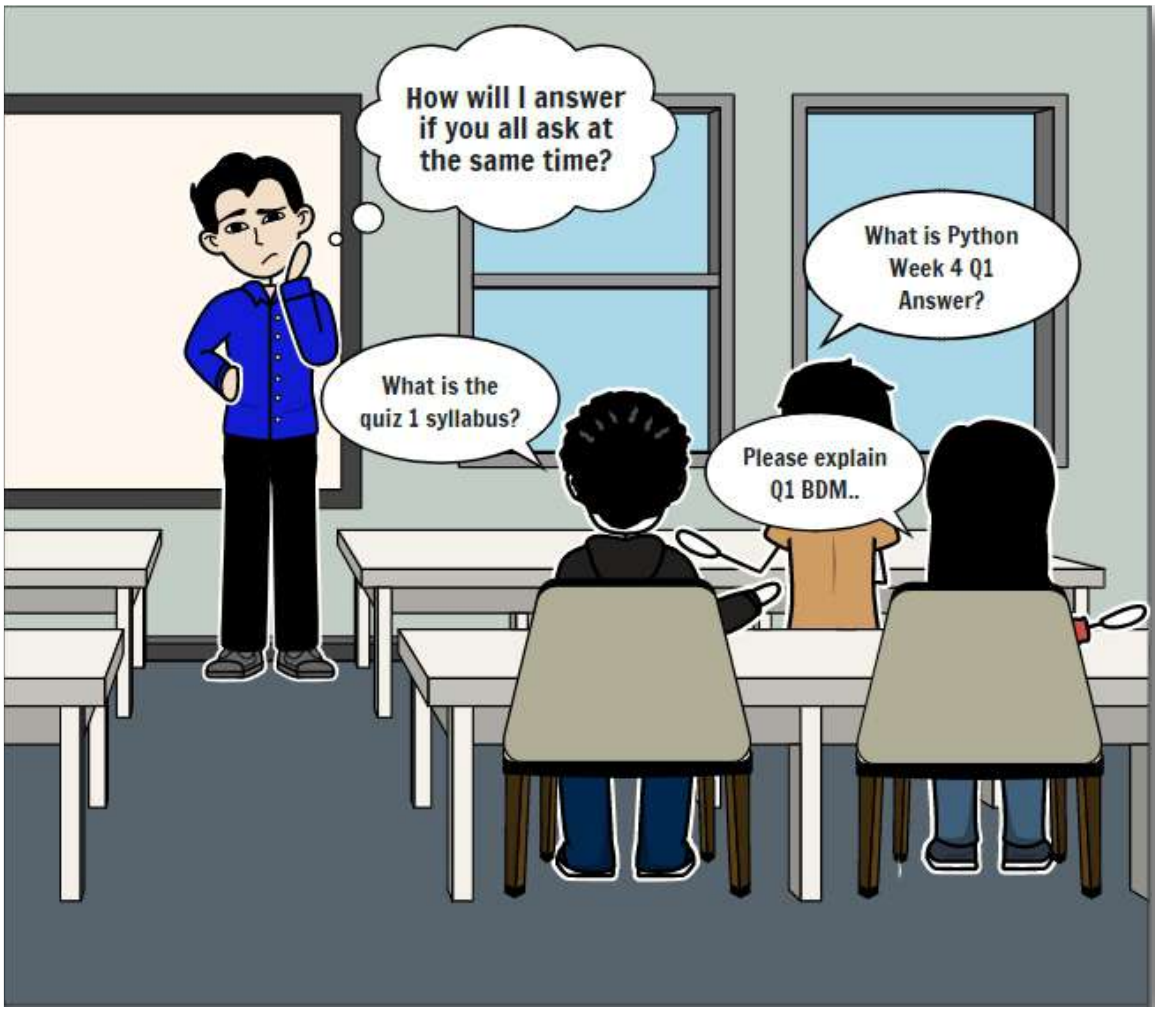
While the “Satisfaction” component addresses the following questions:

- What motivates people to use the system?
- What does it enable people to accomplish?
- What need does the system fill?

Benefits of using Storyboards:

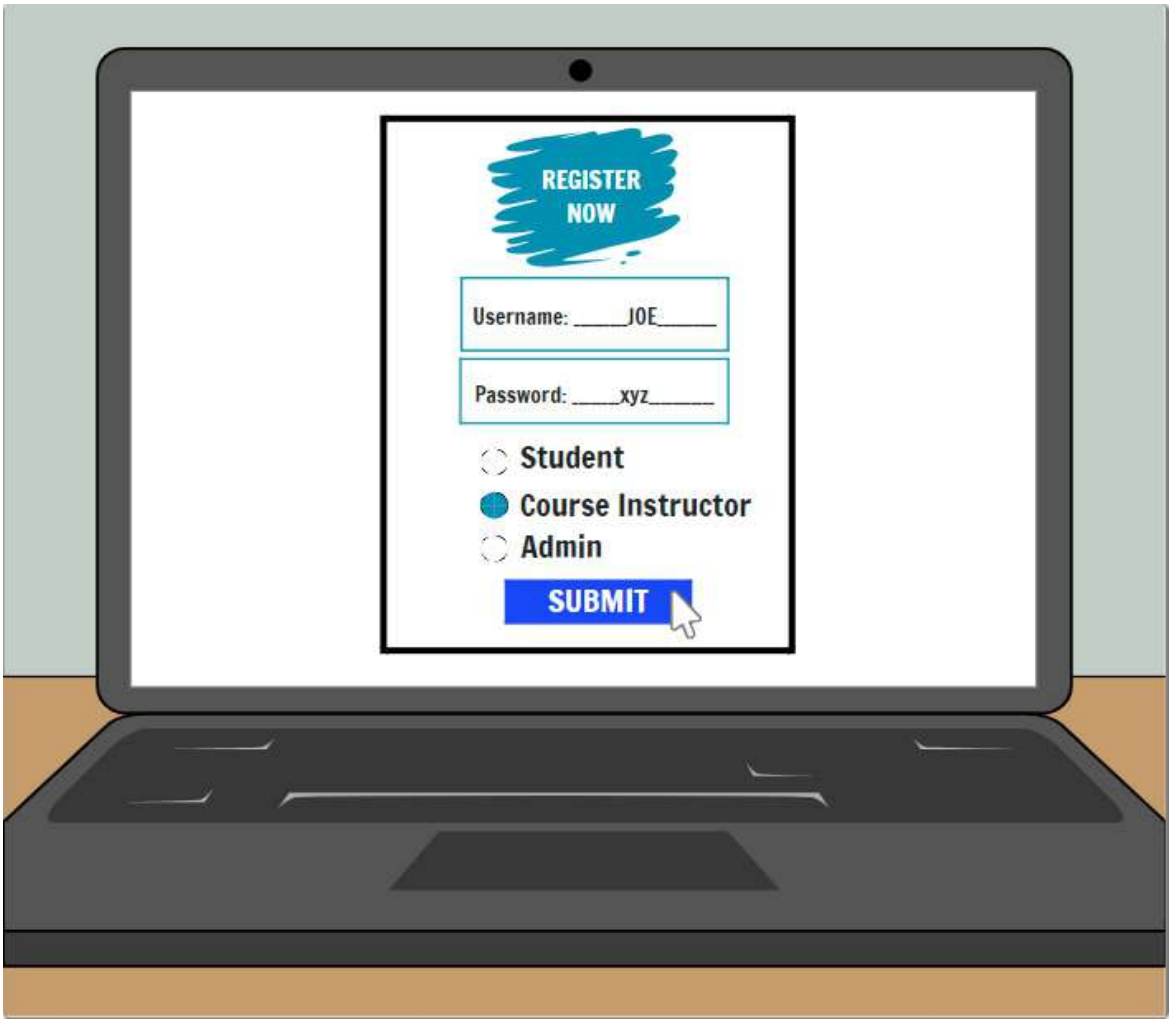
- Emphasizes how interface accomplishes a task
- Avoids commitment to a particular user interface
- Shared understanding among stakeholders.

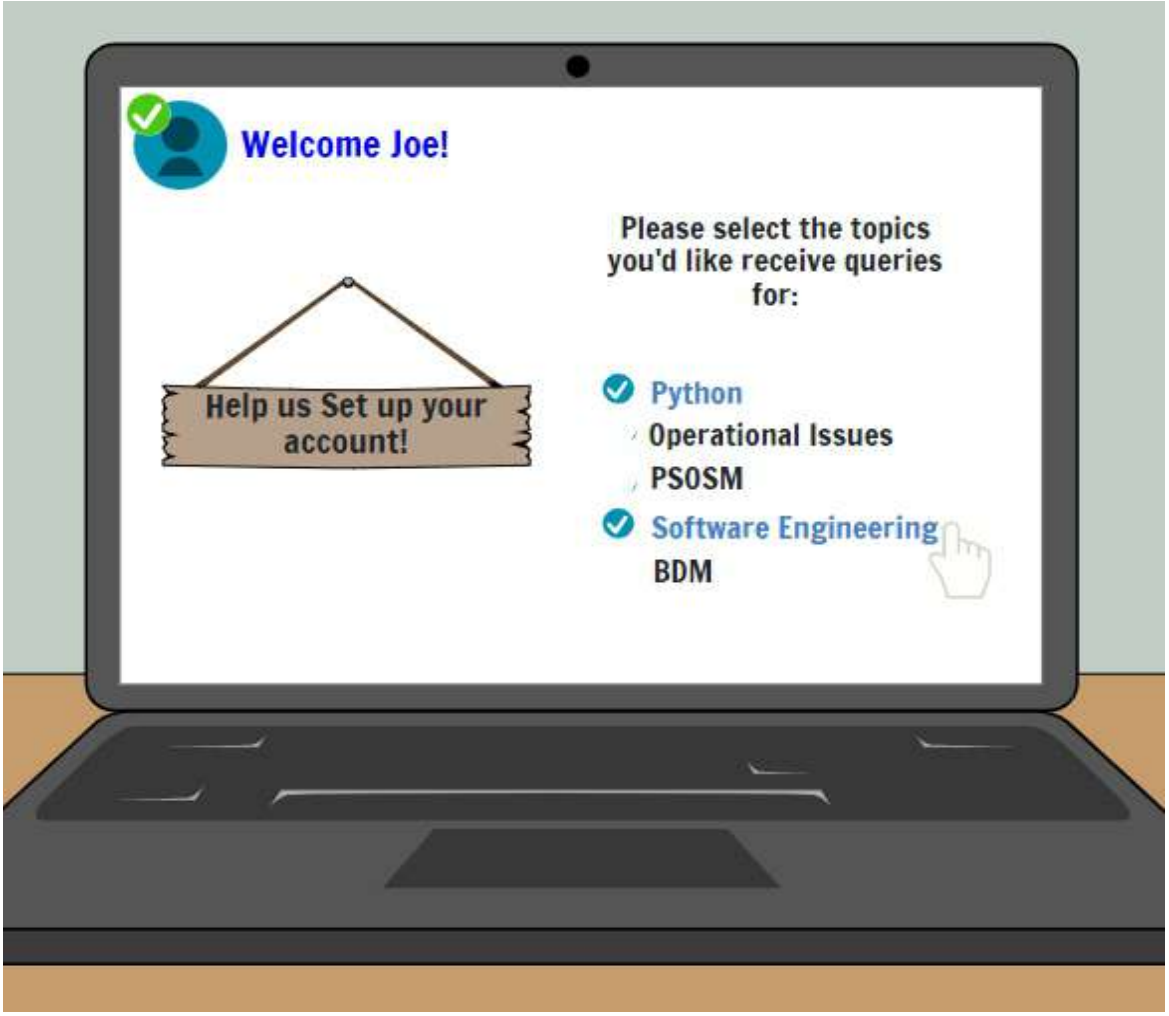
If you have any questions please ask...

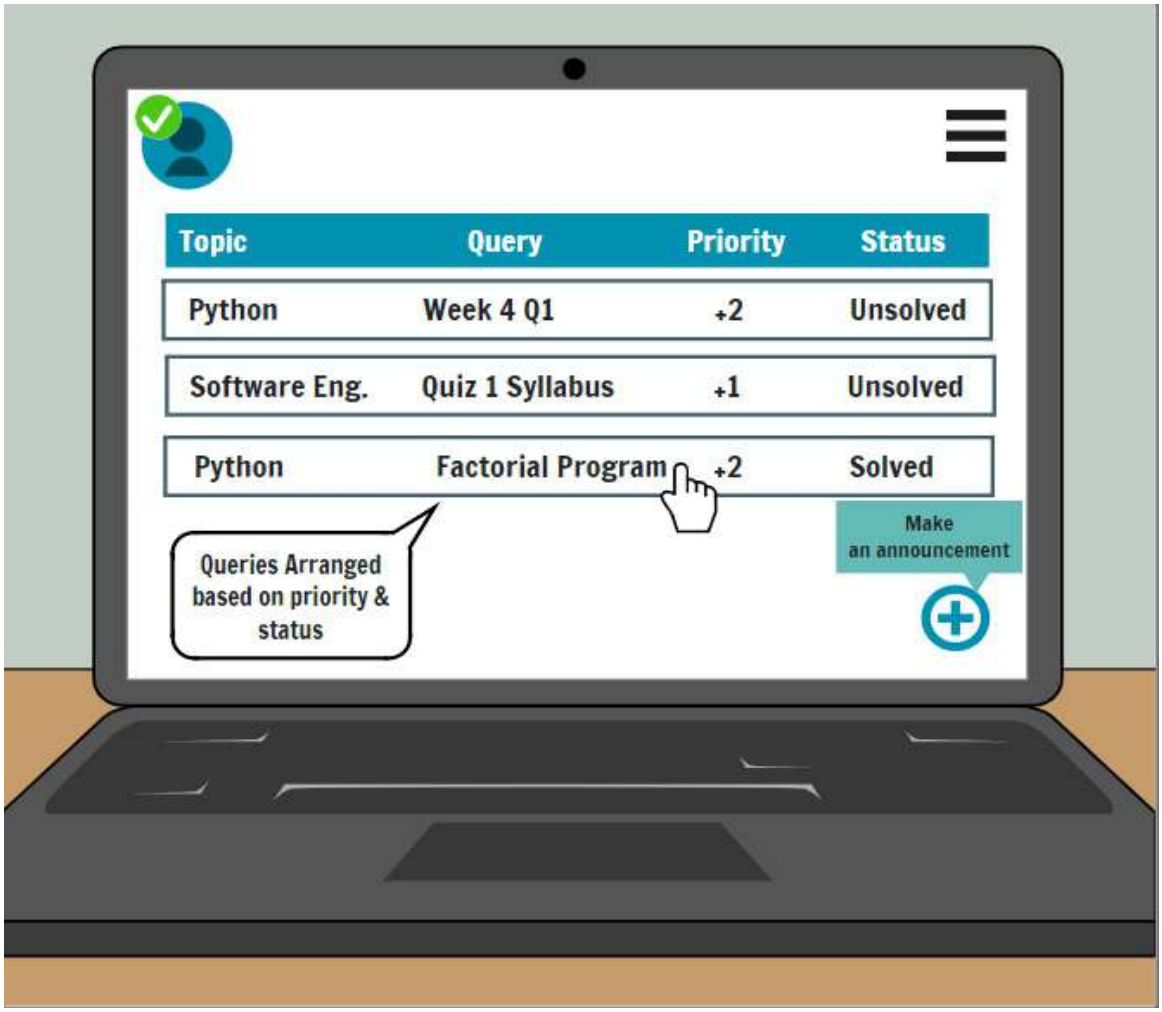

How will I answer if you all ask at the same time?
What is Python Week 4 Q1 Answer?
What is the quiz 1 syllabus?
Please explain Q1 BDM..

Hi Joe! What happened? You are looking confused...
It's really difficult to answer the queries when all the students ask at the same time.. Some even ask the same questions twice...

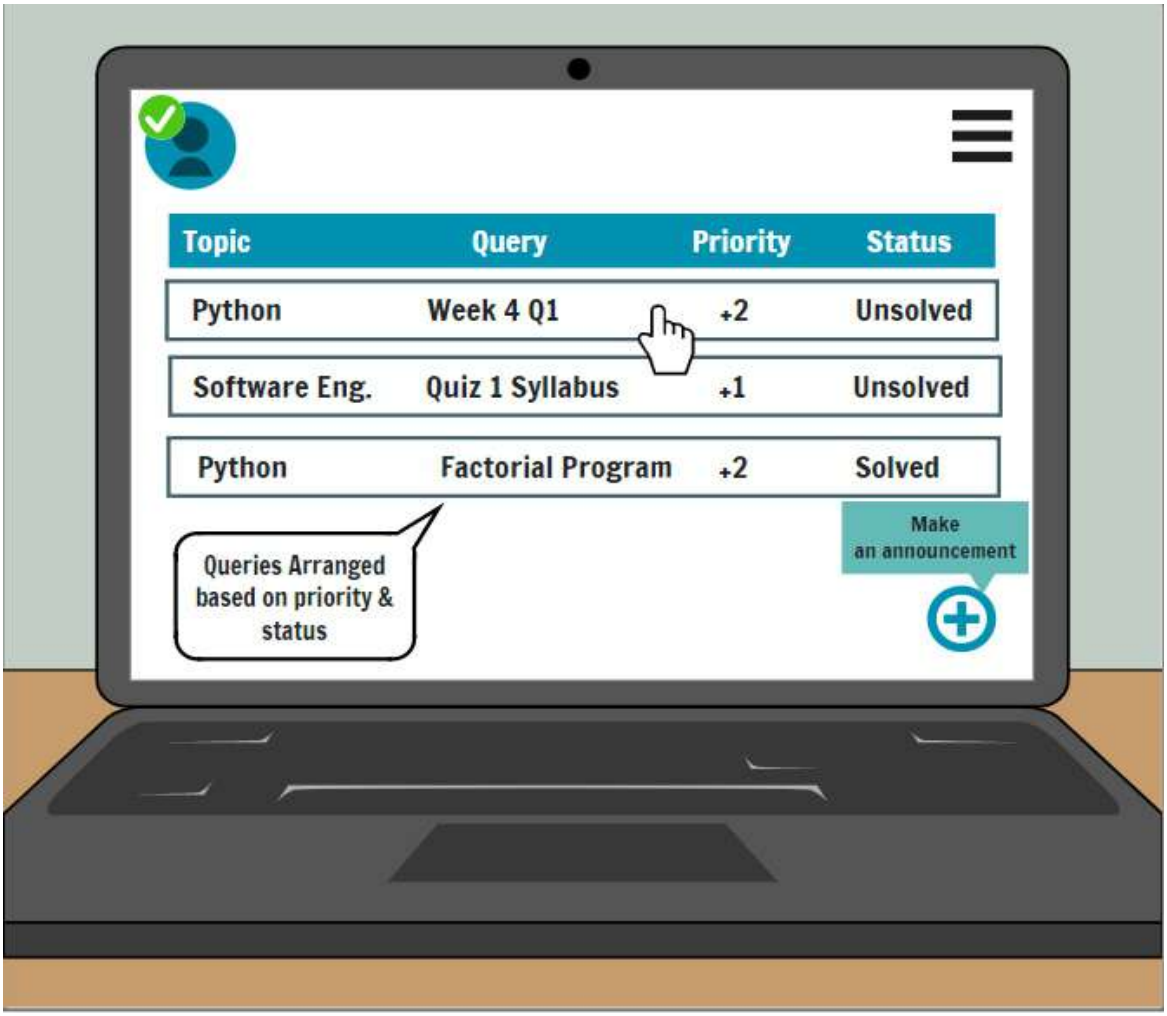
What's that ?
You should try Quick Resolve for effective query addressal.

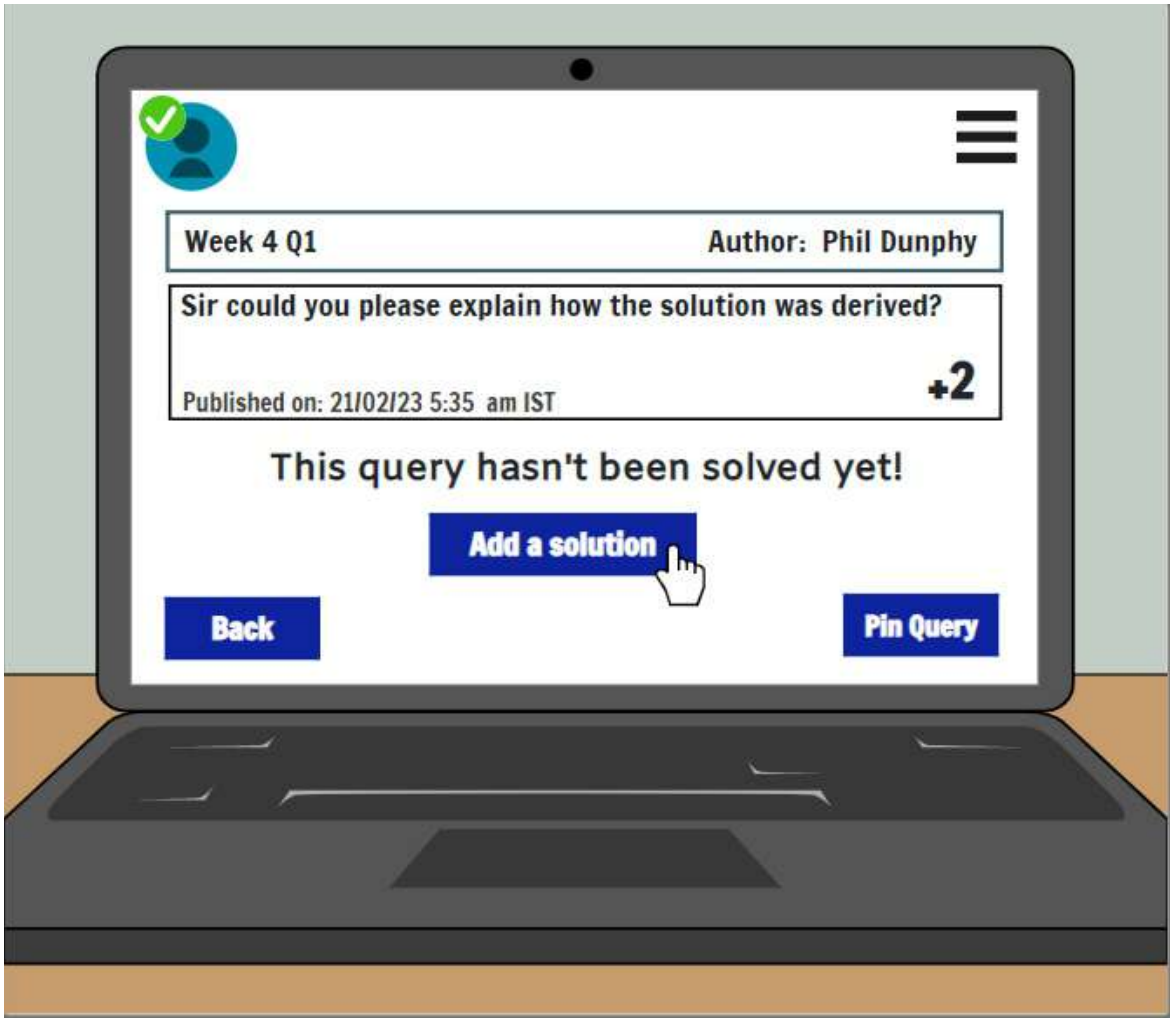

REGISTER NOW
Username: _____JOE_____
Password: _____xyz_____
Student
Course Instructor
Admin
SUBMIT

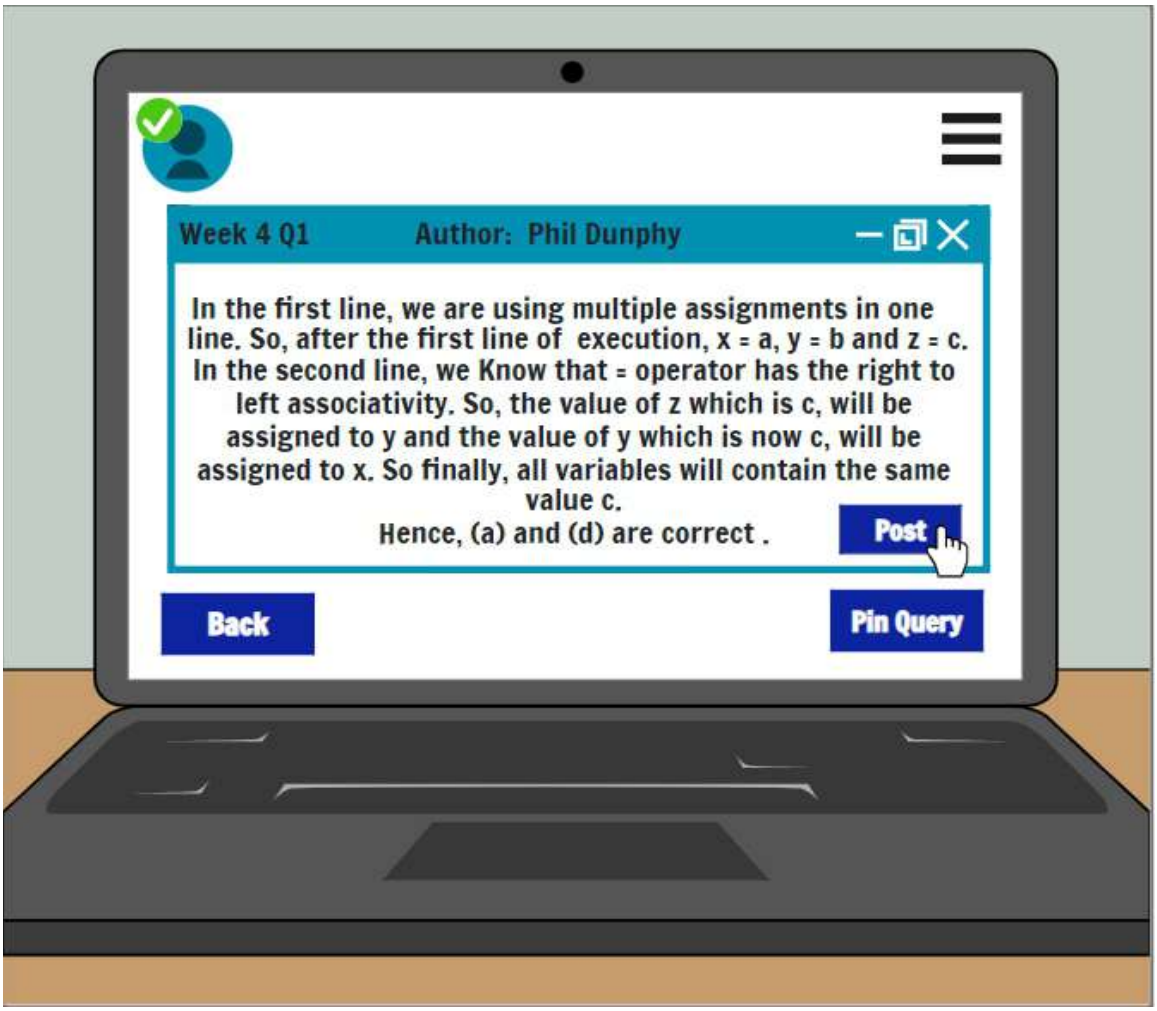

Welcome Joe!
Help us Set up your account!
Please select the topics you'd like receive queries for:
Python
Operational Issues
PSOSM
Software Engineering
BDM


Topic
Query
Priority
Status
Python
Week 4 Q1
+2
Unsolved
Software Eng.
Quiz 1 Syllabus
+1
Unsolved
Python
Factorial Program
+2
Solved
Queries Arranged based on priority & status
Make an announcement

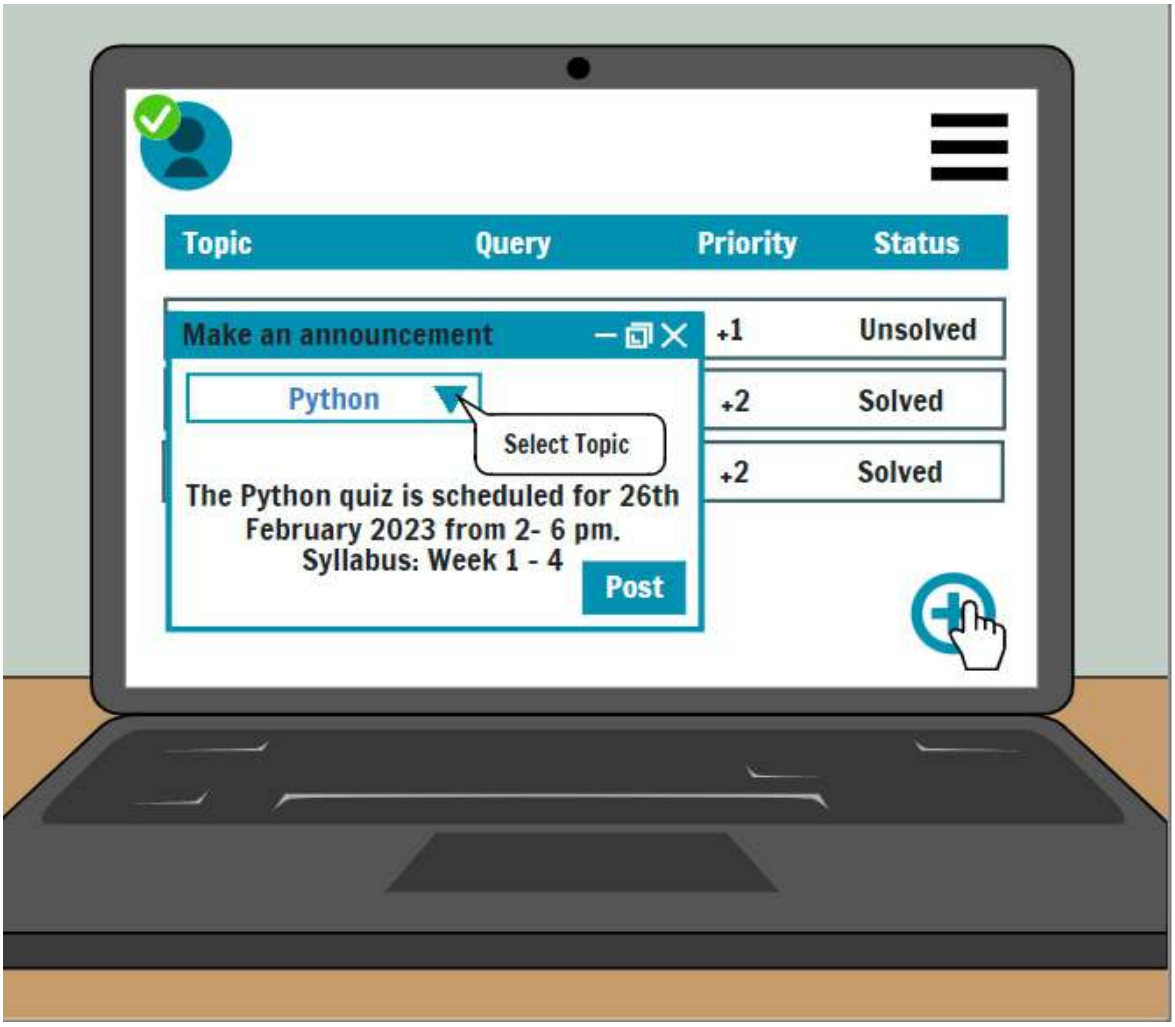
Factorial Program
Author: Jay Pritchett
Sir can you please explain the recursive algo for computing the factorial?
Published on: 20/02/23 8:30 pm IST
+2
Joe ~Course Instructor
Solved on: 20/02/23 9:57 pm IST
Herein, the recursive function will call itself until the value isn't equal to zero. The following formula is used: .....
Back
Pin Query


Topic
Query
Priority
Status
Python
Week 4 Q1
+2
Unsolved
Software Eng.
Quiz 1 Syllabus
+1
Unsolved
Python
Factorial Program
+2
Solved
Queries Arranged based on priority & status
Make an announcement


Week 4 Q1
Author: Phil Dunphy
Sir could you please explain how the solution was derived?
Published on: 21/02/23 5:35 am IST
+2
This query hasn't been solved yet!
Add a solution
Back
Pin Query

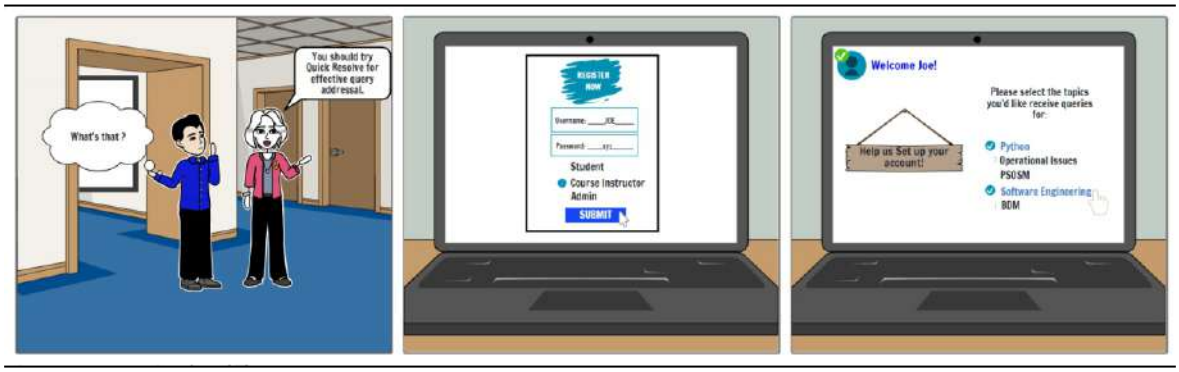

Week 4 Q1
Author: Phil Dunphy
In the first line, we are using multiple assignments in one line. So, after the first line of execution, x = a, y = b and z = c. In the second line, we Know that = operator has the right to left associativity. So, the value of z which is c, will be assigned to y and the value of y which is now c, will be assigned to x. So finally, all variables will contain the same value c.
Hence, (a) and (d) are correct .
Post
Back
Pin Query

Week 4 Q1
Author: Phil Dunphy
Sir could you please explain how the solution was derived?
+2
Published on: 21/02/23 5:35 am IST
Joe ~Course Instructor
Solved on: 21/02/23 12:03 pm IST
In the first line, we are using multiple assignments in one line. So, after the first line.....
Back
Pin Query


Topic
Query
Priority
Status
Make an announcement
+1
Unsolved
Python
Select Topic
+2
Solved
+2
Solved
The Python quiz is scheduled for 26th February 2023 from 2- 6 pm.
Syllabus: Week 1 - 4
Post

Wow that's such an easy and efficient way to address student queries!
Quick Resolve

If you have any questions please ask...
How will I answer if you all ask at the same time?
What is the quiz 1 syllabus?
What is Python Week 4 Q1 Answer?
Please explain Q1 BDM..
It's really difficult to answer the queries when all the students ask at the same time.. Some even ask the same questions twice...
Hi Joe! What happened? You are looking confused...

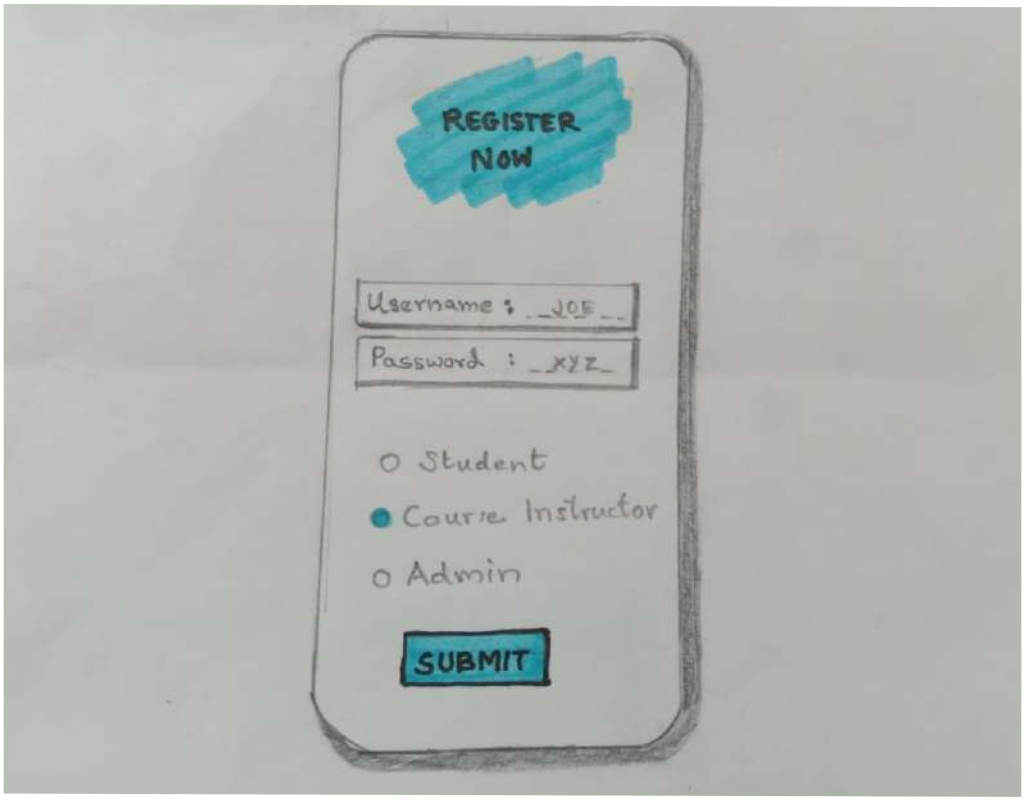

What's that ?
You should try Quick Resolve for effective query addressal.
REGISTER NOW
Student
Course Instructor
Admin
SUBMIT
Welcome Joe!
Please select the topics you'd like receive queries for:
Help us Set up your account!
Python
Operational Issues
PSOSM
Software Engineering
BDM

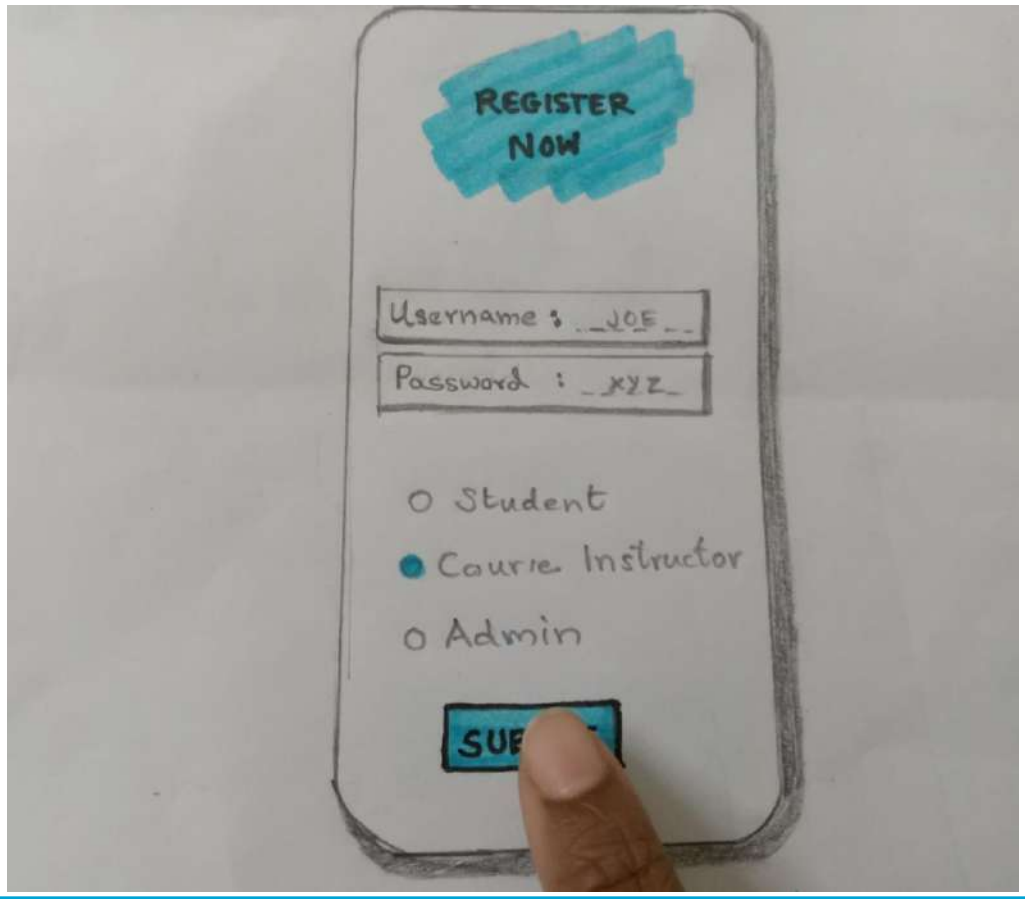
Topic
Query
Priority
Status
Python
Week 4 Q1
+2
Unsolved
Software Eng.
Quiz 1 Syllabus
+1
Unsolved
Python
Factorial Program
+2
Solved
Queries Arranged based on priority & status
Factorial Program
Author: Jay Pritchett
Sir can you please explain the recursive algo for computing the factorial?
+2
Joe - Course Instructor
Herein, the recursive function will call itself until the value isn't equal to zero. The following formula is used:
Back
Pin Query

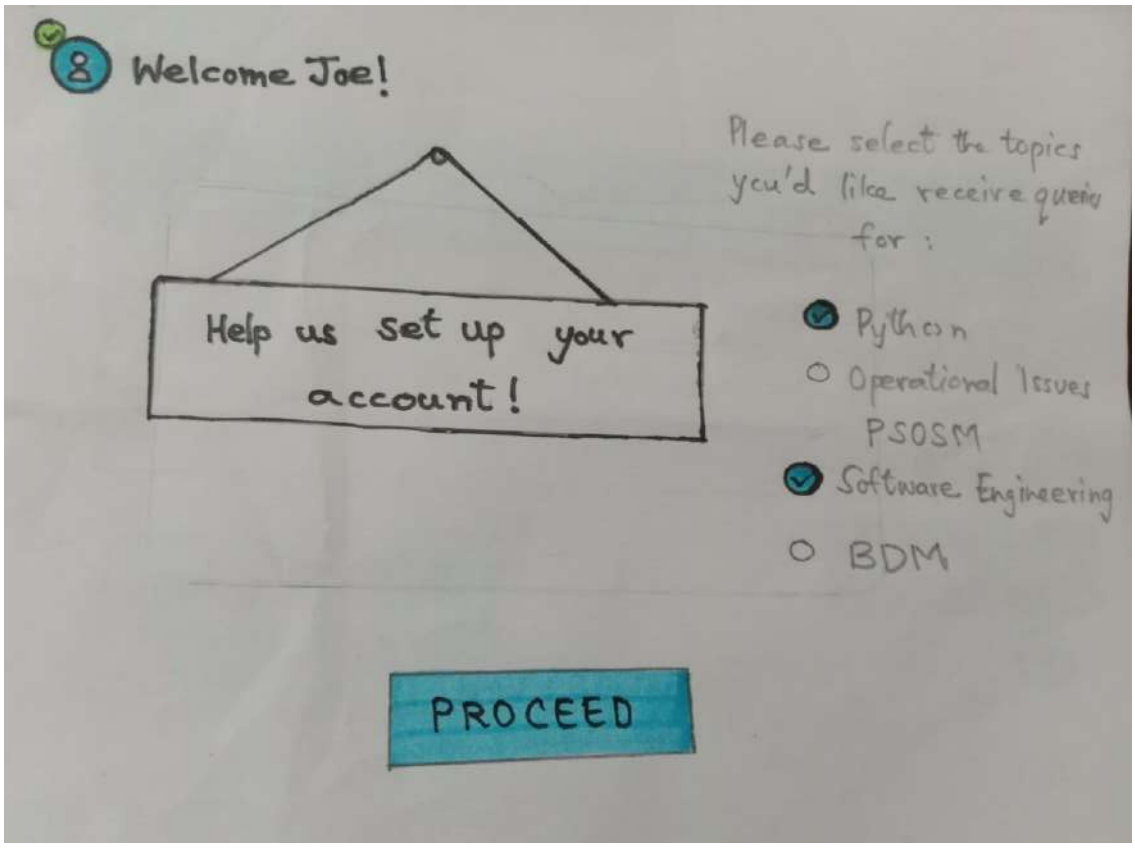
Week 4 Q1
Author: Phil Dunphy
Sir could you please explain how the solution was derived?
+2
This query hasn't been solved yet!
Add a solution
Back
Pin Query
Week 4 Q1
Author: Phil Dunphy
In the first line, we are using multiple assignments in one line. So, after the first line of execution, x = a, y = b and z = c. In the second line, we know that = operator has the right to left associativity. So, the value of z which is c, will be assigned to y and the value of y which is now c, will be assigned to x. So finally, all variables will contain the same value c.
Hence, (a) and (d) are correct.
Post
Week 4 Q1
Author: Phil Dunphy
Sir could you please explain how the solution was derived?
+2
Joe - Course Instructor
In the first line, we are using multiple assignments in one line. So, after the first line.....

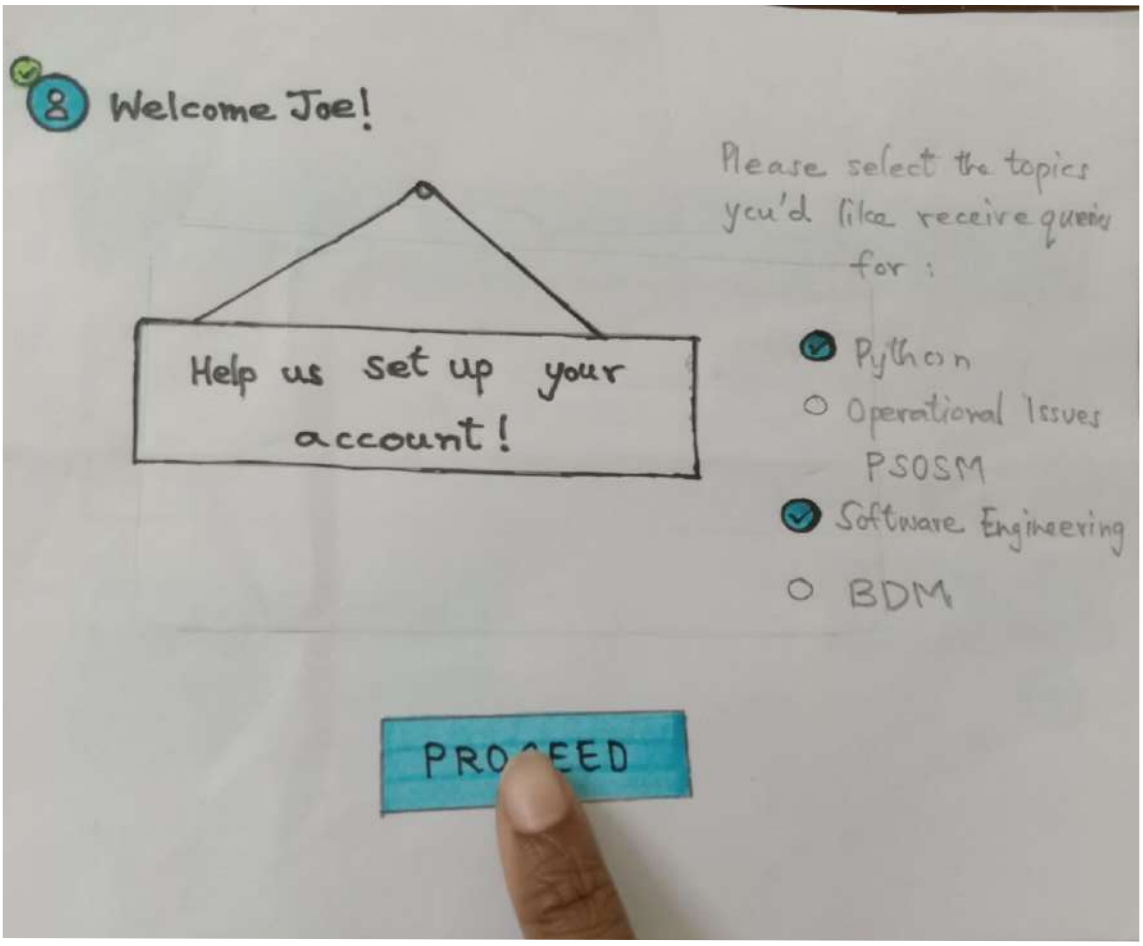
Topic
Query
Priority
Status
Make an announcement
+1
Unsolved
Python
Select Topic
+2
Solved
+2
Solved
The Python quiz is scheduled for 26th February 2023 from 2- 6 pm.
Syllabus: Week 1 - 4
Post
Wow that's such an easy and efficient way to address student queries!
Quick Resolve

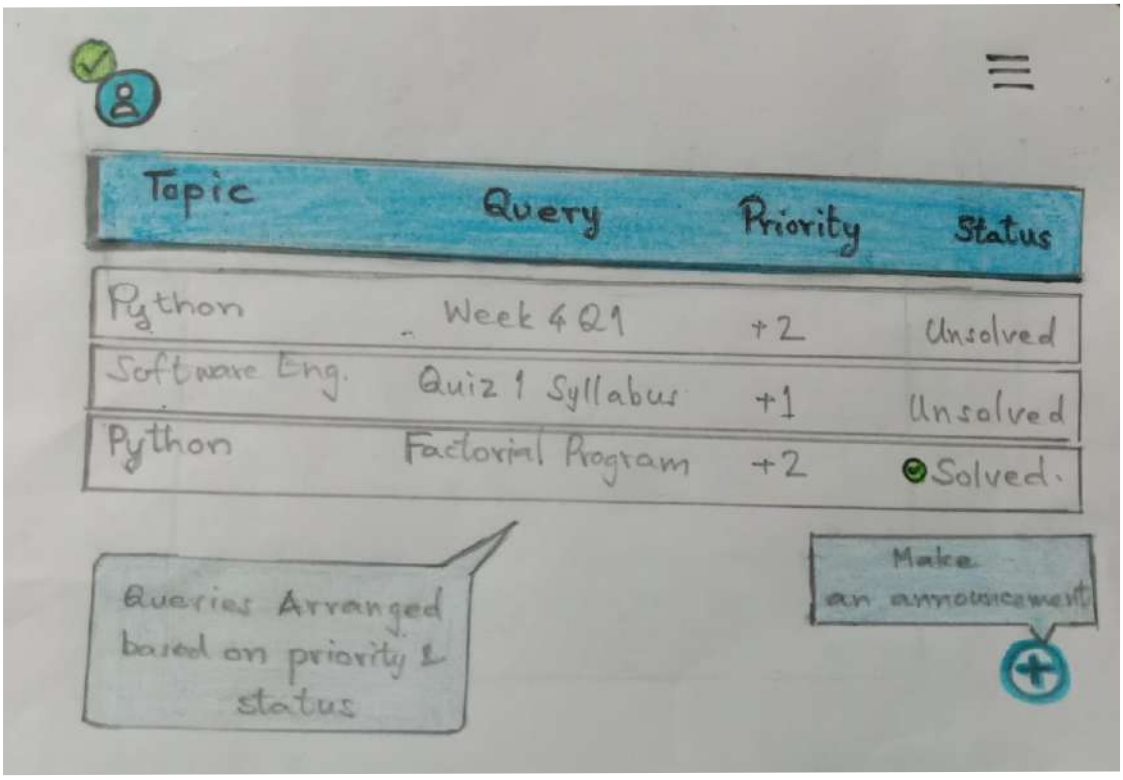
Paper Prototype

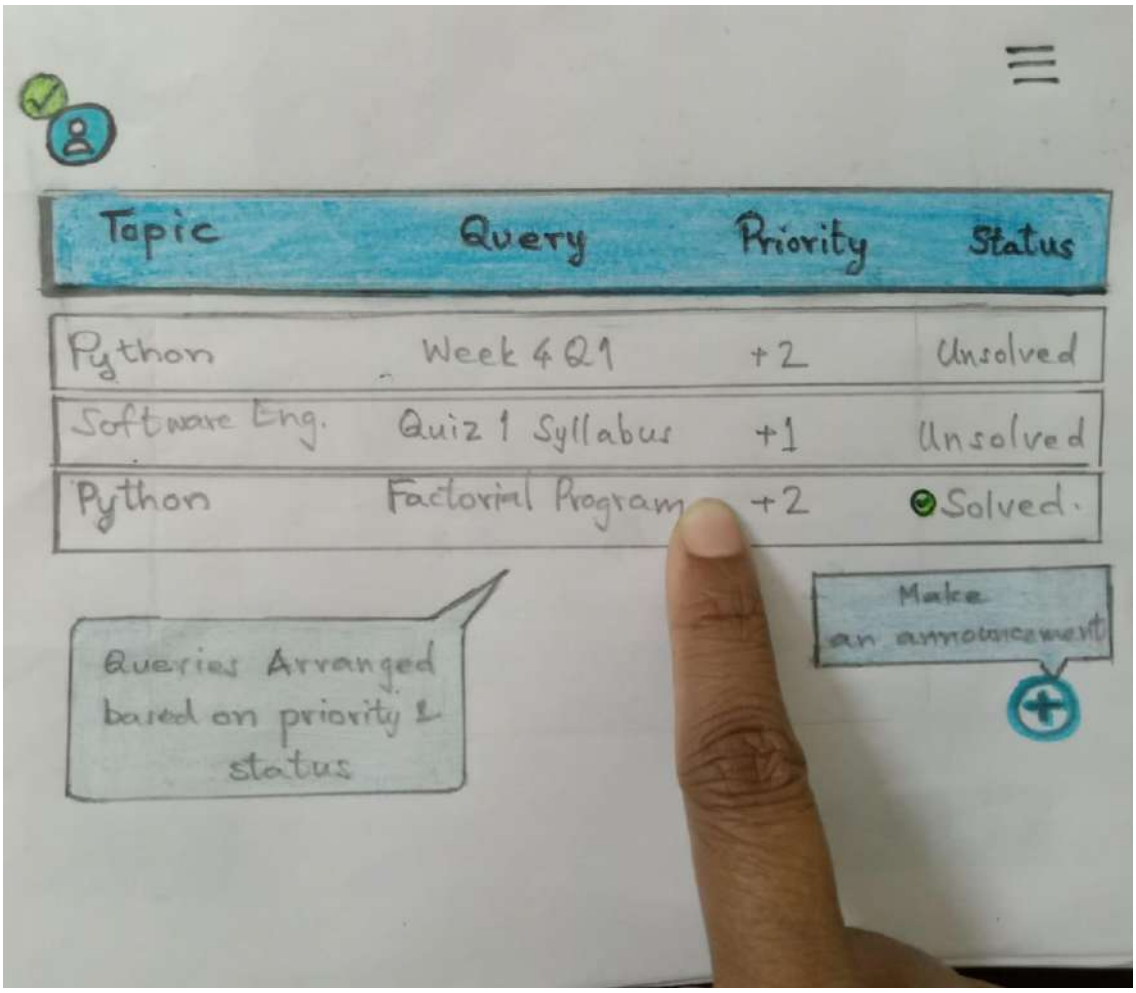


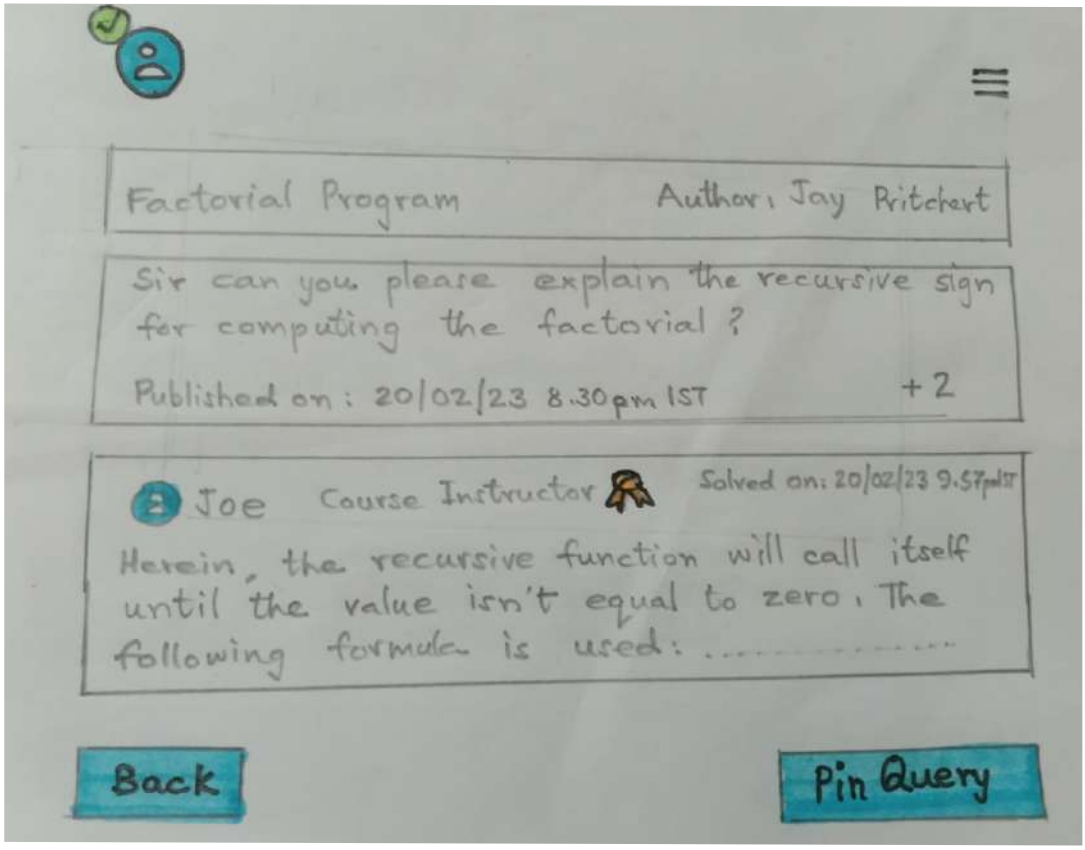


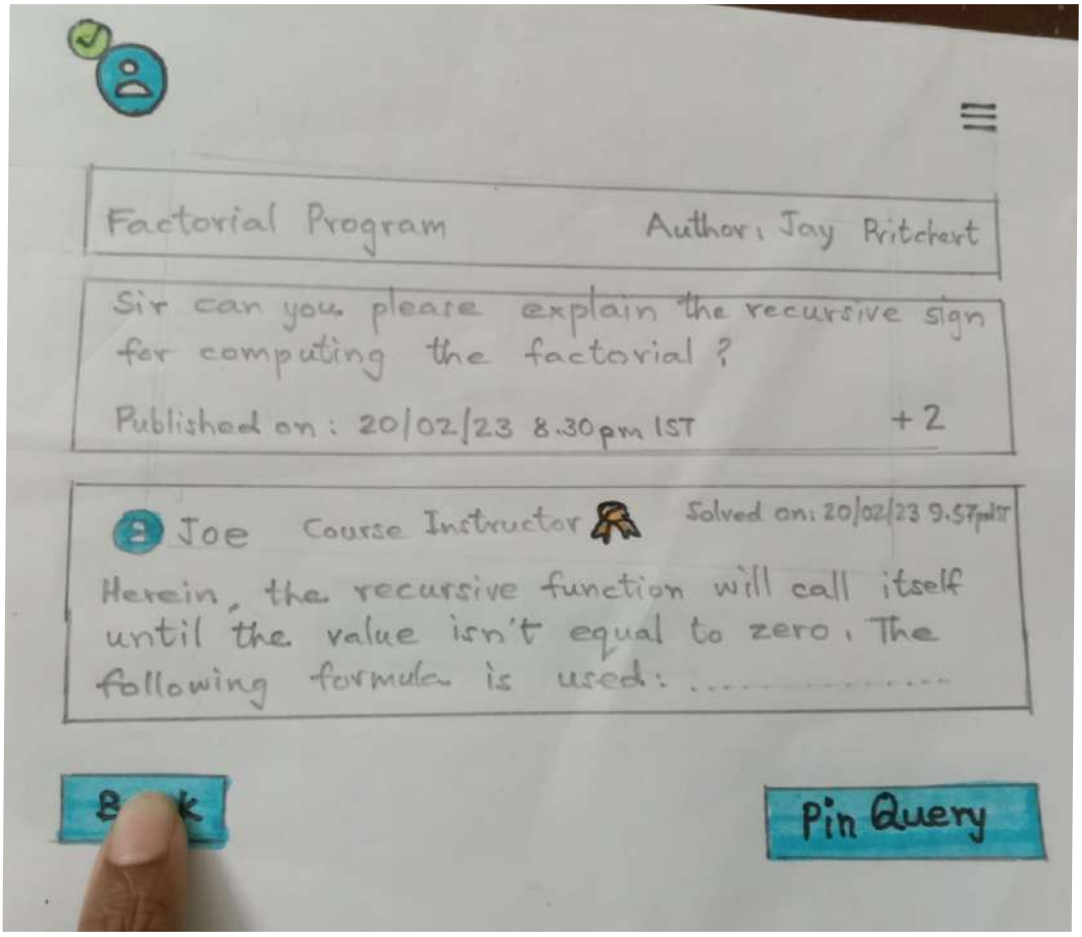

Welcome Joe!
Help us set up your account!
Please select the topics you'd like receive queries for :
Python
Operational Issues PSOSM
Software Engineering
BDM
PROCEED

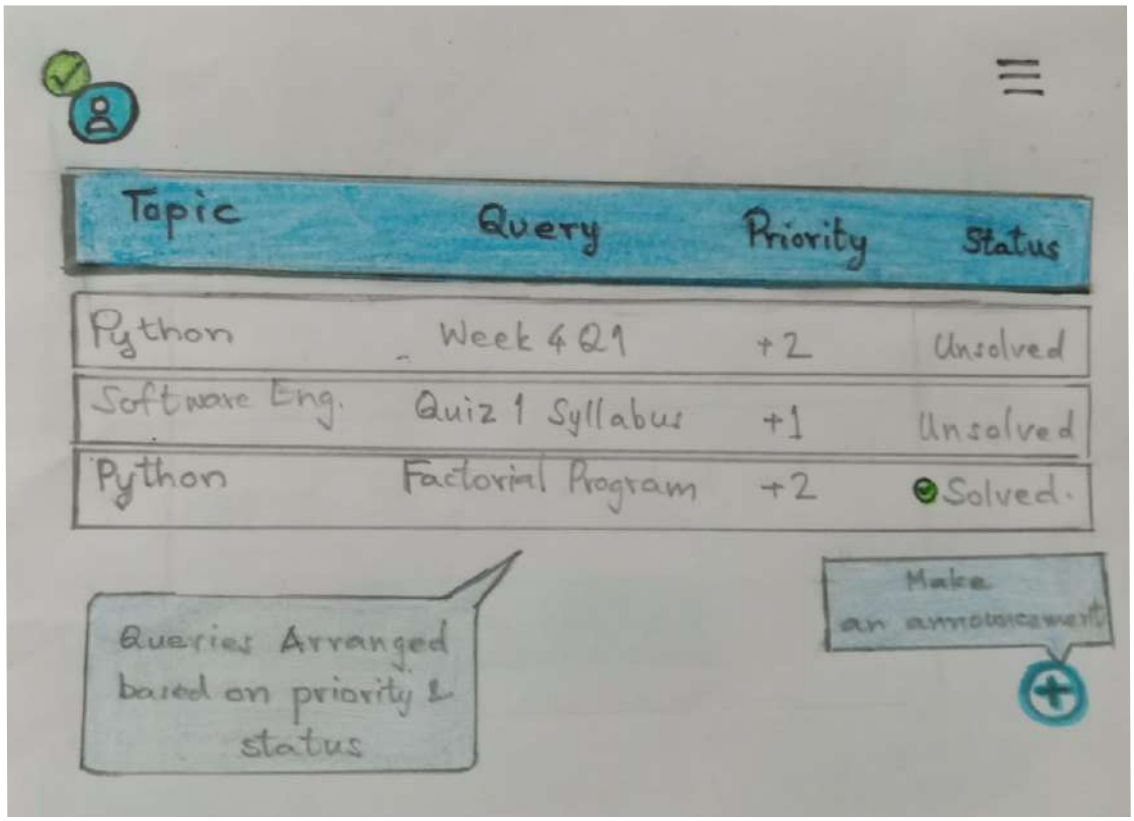

Welcome Joe!
Help us set up your account!
Please select the topics you'd like receive queries for :
Python
Operational Issues PSOSM
Software Engineering
BDM
PROCEED

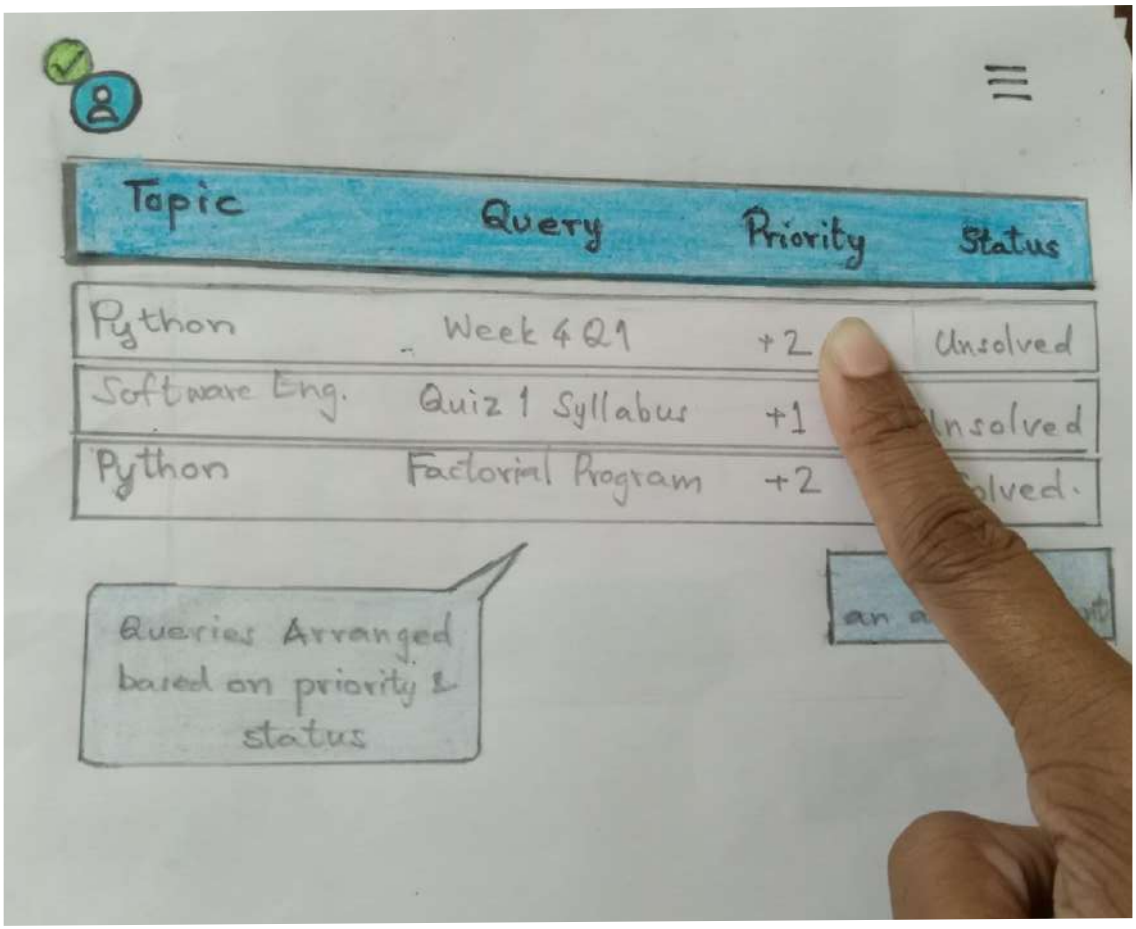

Topic
Query
Priority
Status
Python
Week 4 Q1
+2
Unsolved
Software Eng.
Quiz 1 Syllabus
+1
Unsolved
Python
Factorial Program
+2
Solved.
Queries Arranged based on priority & status
Make an announcement

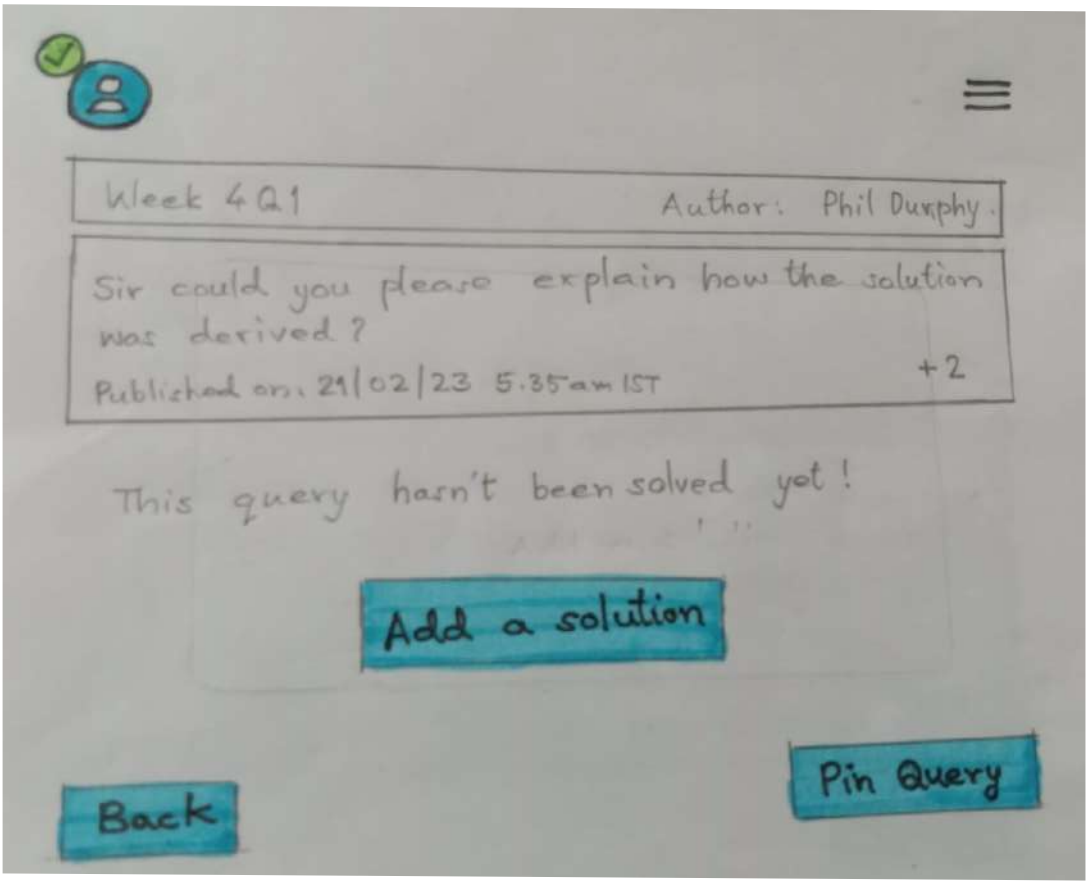

Topic
Query
Priority
Status
Python
Week 4 Q1
+2
Unsolved
Software Eng.
Quiz 1 Syllabus
+1
Unsolved
Python
Factorial Program
+2
Solved.
Queries Arranged based on priority & status
Make an announcement

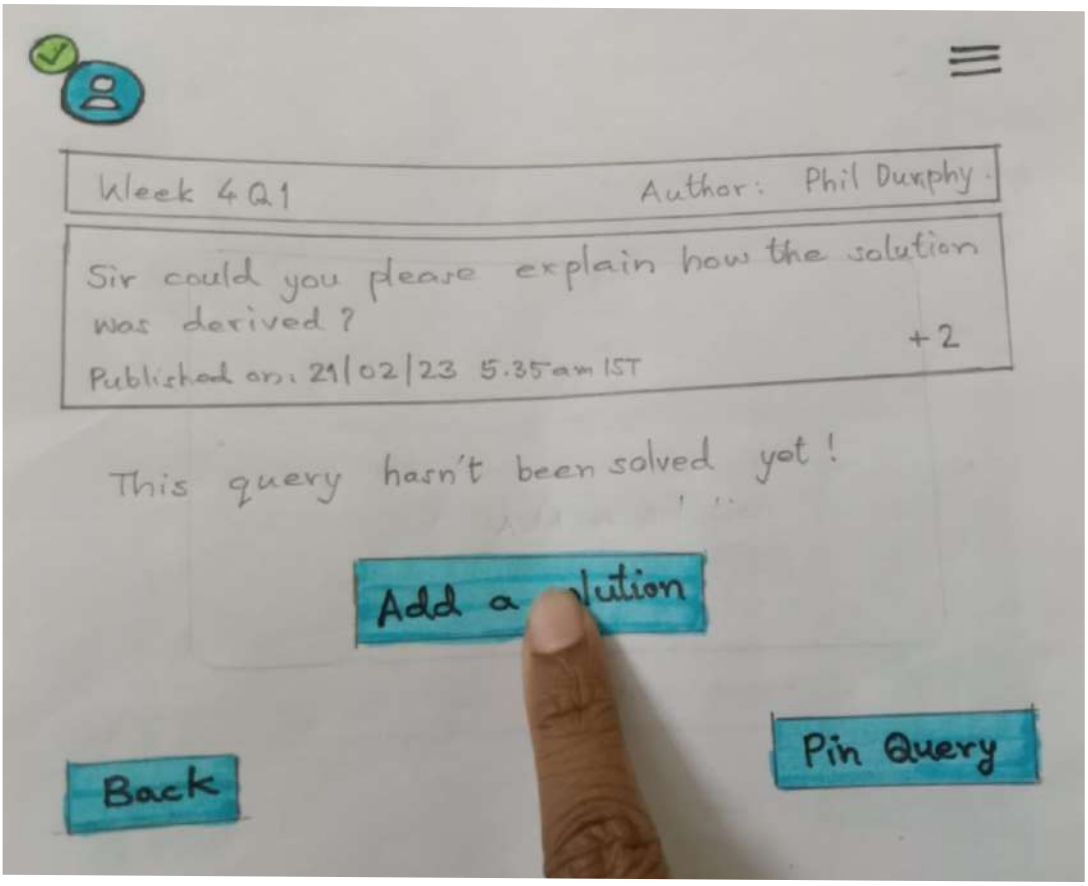

Factorial Program
Author: Jay Pritchert
Sir can you please explain the recursive sign for computing the factorial ?
Published on: 20/02/23 8.30pm IST
+2
Joe
Course Instructor
Solved on: 20/02/23 9.57pm IST
Herein, the recursive function will call itself until the value isn't equal to zero. The following formula is used: .....
Back
Pin Query

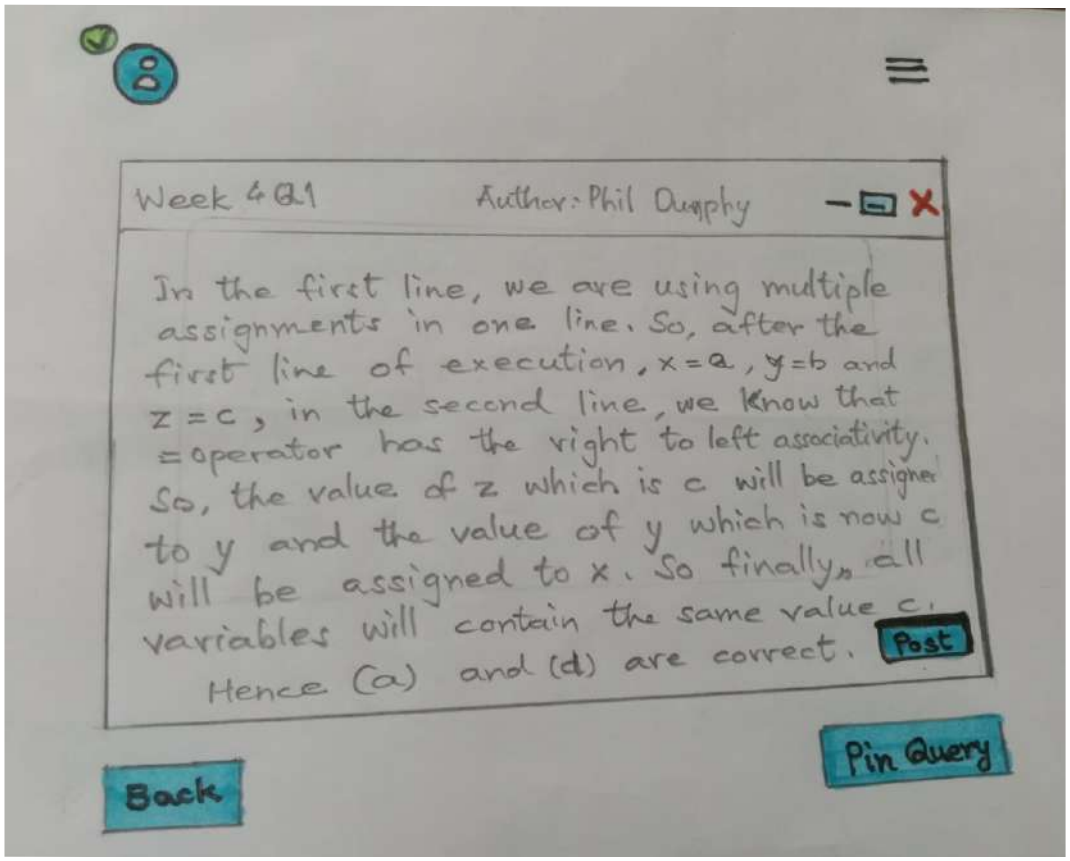

Factorial Program
Author: Jay Pritchert
Sir can you please explain the recursive sign for computing the factorial ?
Published on: 20/02/23 8.30pm IST
+2
Joe
Course Instructor
Solved on: 20/02/23 9.57pm IST
Herein, the recursive function will call itself until the value isn't equal to zero. The following formula is used: .....
Pin Query

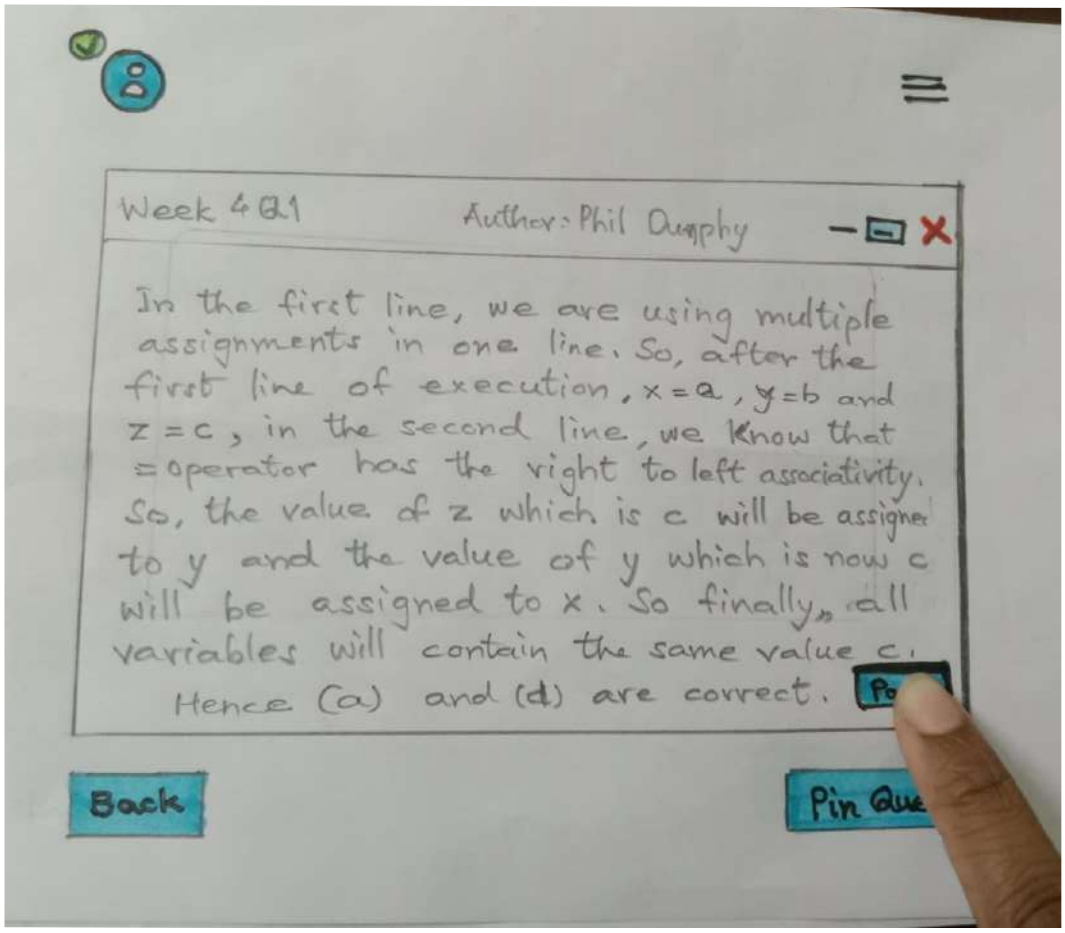

Topic
Query
Priority
Status
Python
Week 4 Q1
+2
Unsolved
Software Eng.
Quiz 1 Syllabus
+1
Unsolved
Python
Factorial Program
+2
Solved.
Queries Arranged based on priority & status
Make an announcement

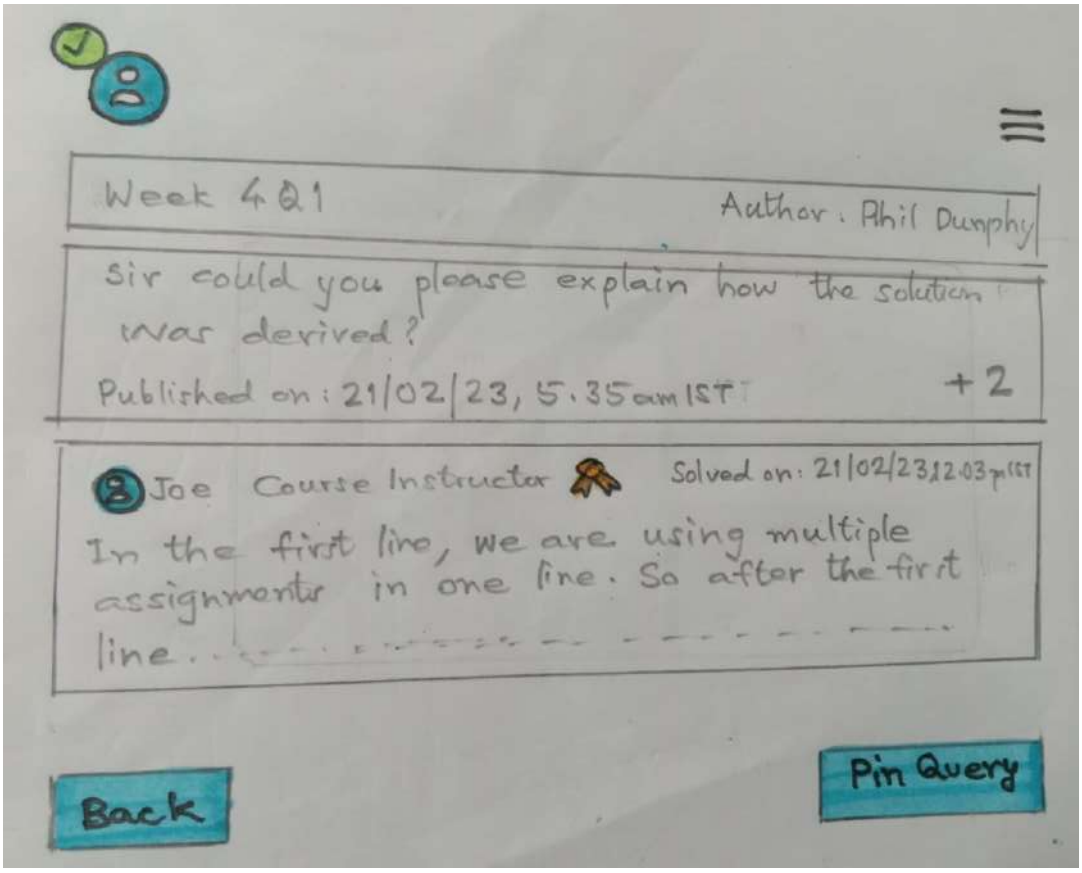

Topic
Query
Priority
Status
Python
Week 4 Q1
+2
Unsolved
Software Eng.
Quiz 1 Syllabus
+1
Python
Factorial Program
+2
Queries Arranged based on priority & status

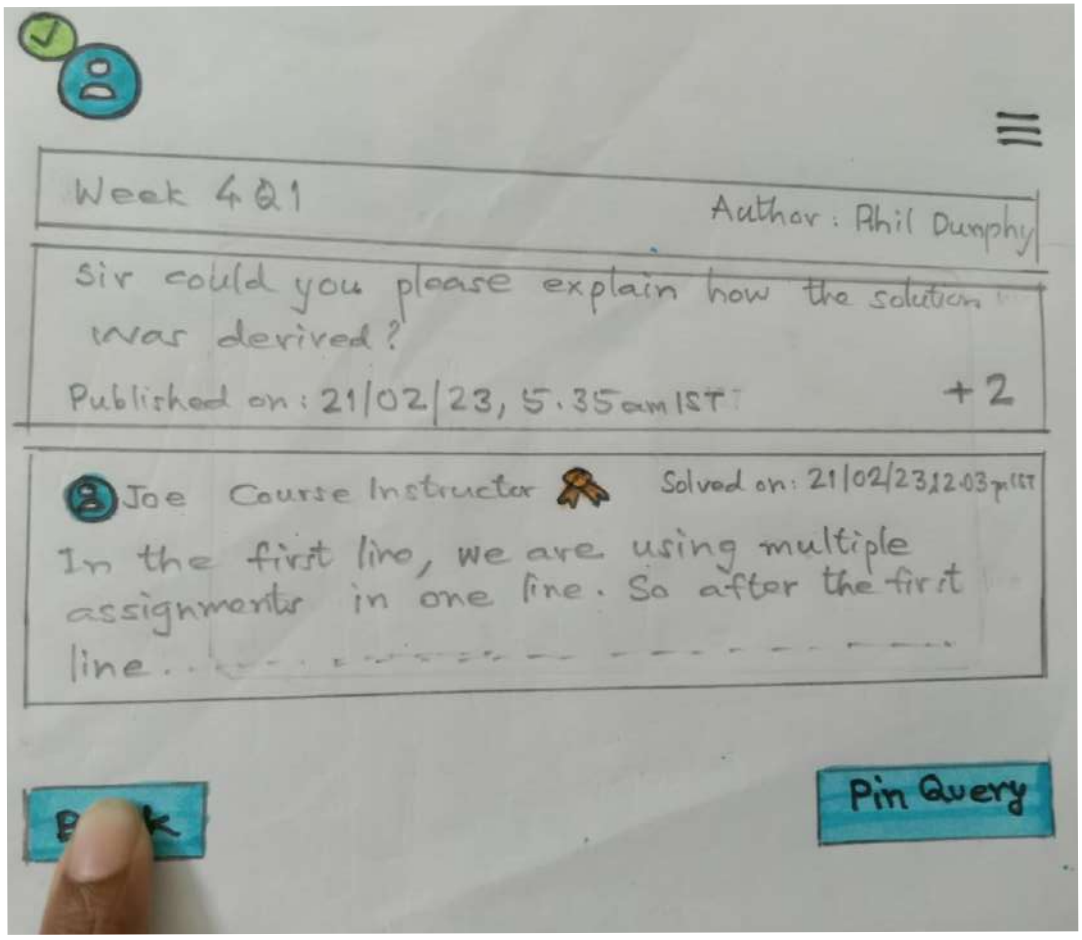

Week 4 Q1
Author: Phil Durphy.
Sir could you please explain how the solution was derived ?
Published on: 21/02/23 5.35am IST
+2
This query hasn't been solved yet !
Add a solution
Pin Query
Back

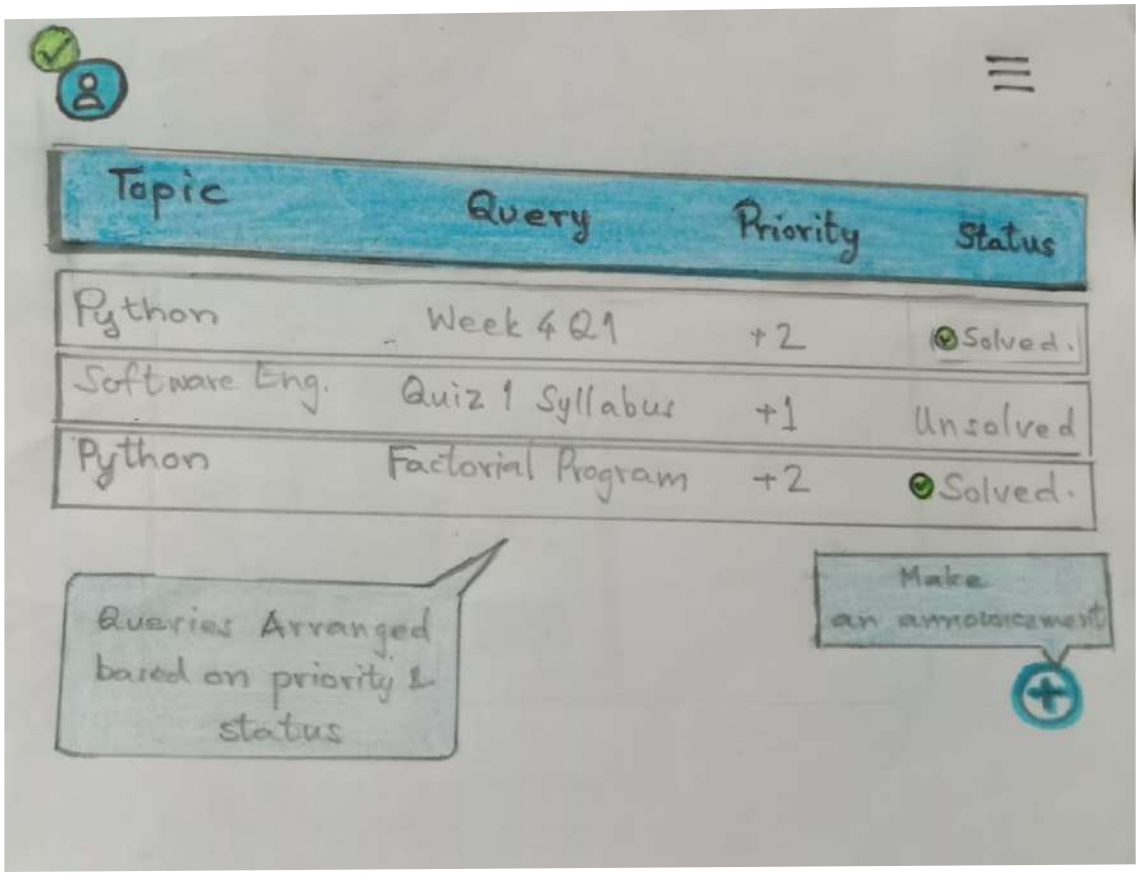

Week 4 Q1
Author: Phil Durphy.
Sir could you please explain how the solution was derived ?
Published on: 21/02/23 5.35am IST
+2
This query hasn't been solved yet !
Add a solution
Pin Query
Back

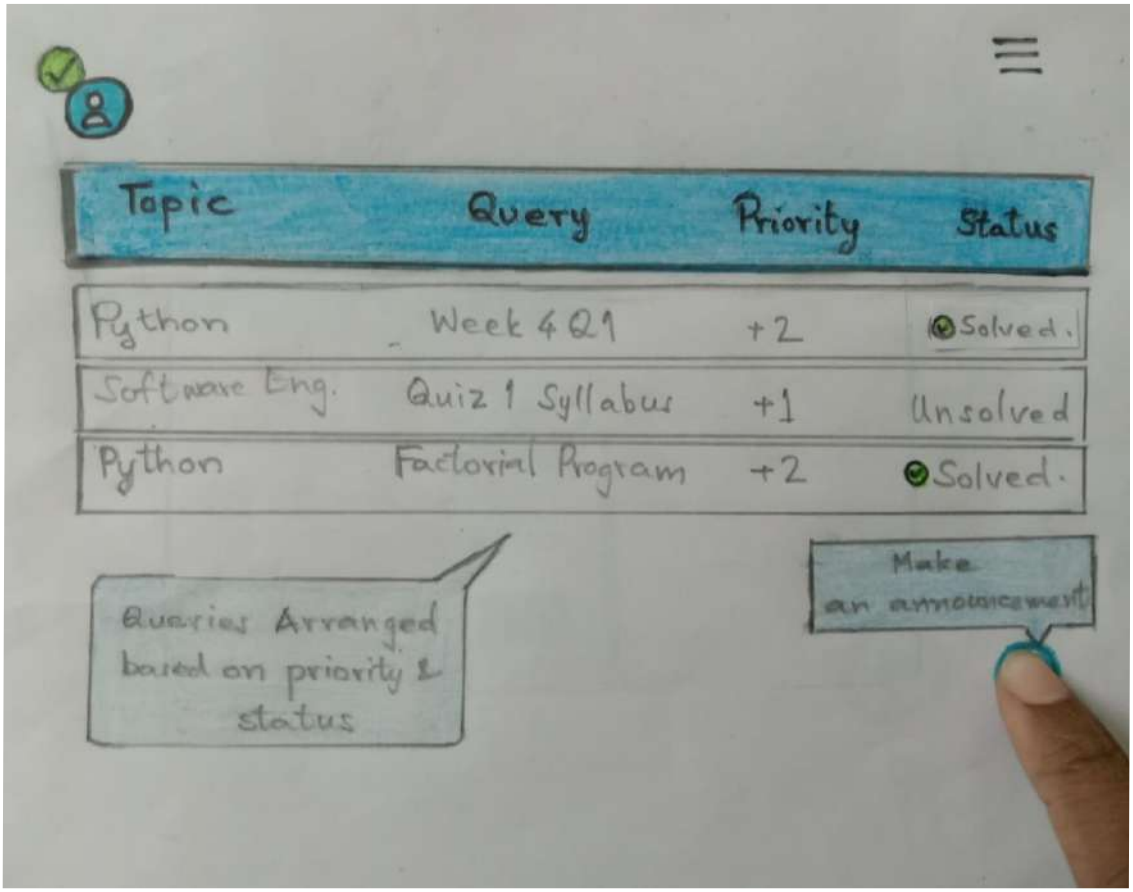

Week 4 Q1
Author: Phil Dunphy
In the first line, we are using multiple assignments in one line. So, after the first line of execution, x = a, y = b and z = c, in the second line, we know that = operator has the right to left associativity. So, the value of z which is c will be assigned to y and the value of y which is now c will be assigned to x. So finally, all variables will contain the same value c.
Hence (a) and (d) are correct.
Post
Back
Pin Query

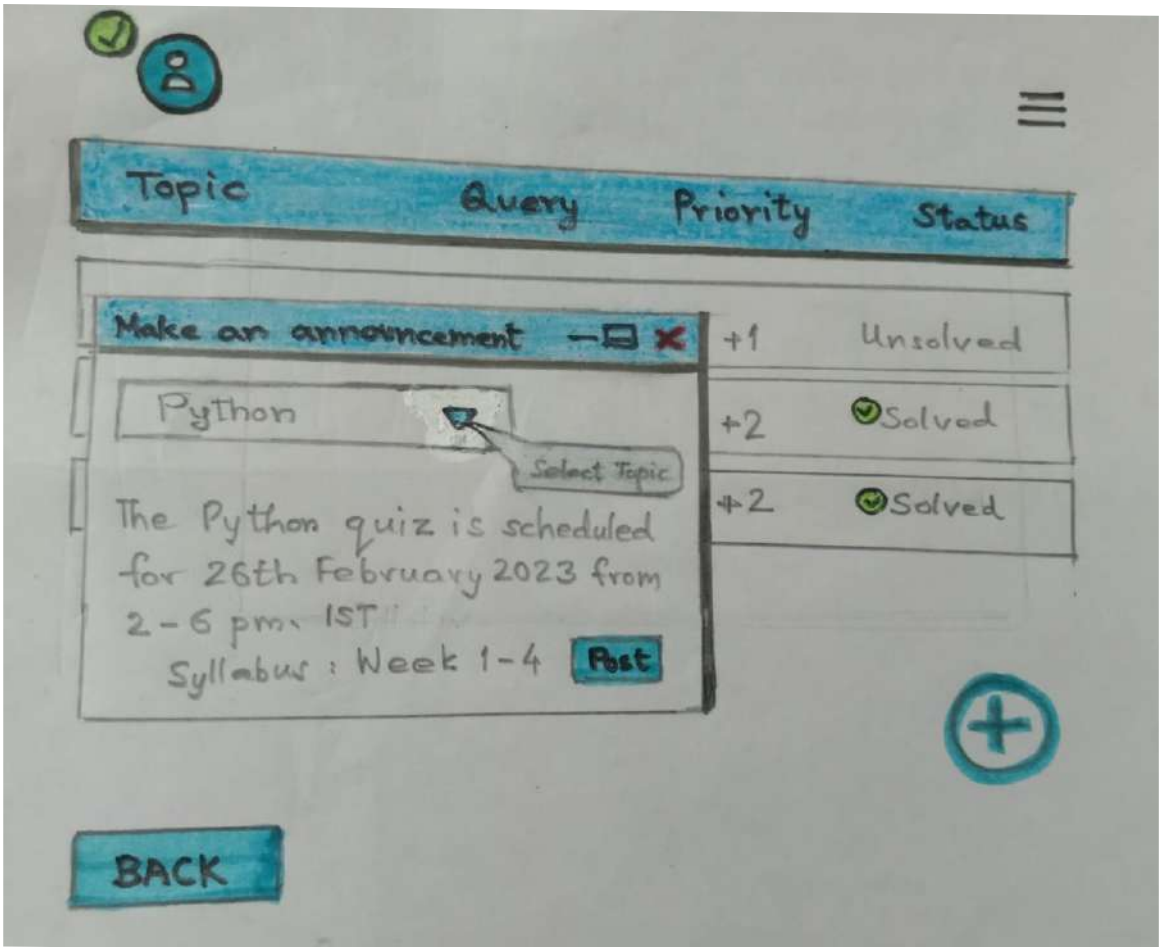

Week 4 Q1
Author: Phil Dunphy
In the first line, we are using multiple assignments in one line. So, after the first line of execution, x = a, y = b and z = c, in the second line, we know that = operator has the right to left associativity. So, the value of z which is c will be assigned to y and the value of y which is now c will be assigned to x. So finally, all variables will contain the same value c.
Hence (a) and (d) are correct.
Back


Week 4Q1
Author: Phil Dunphy
Sir could you please explain how the solution was derived?
Published on: 21/02/23, 5.35am IST
+2
Joe Course Instructor
Solved on: 21/02/23 12.03pm IST
In the first line, we are using multiple assignments in one line. So after the first line.
Back
Pin Query


Week 4Q1
Author: Phil Dunphy
Sir could you please explain how the solution was derived?
Published on: 21/02/23, 5.35am IST
+2
Joe Course Instructor
Solved on: 21/02/23 12.03pm IST
In the first line, we are using multiple assignments in one line. So after the first line.
Pin Query


Topic
Query
Priority
Status
Python
Week 4 Q1
+2
Solved.
Software Eng.
Quiz 1 Syllabus
+1
Unsolved
Python
Factorial Program
+2
Solved.
Queries Arranged based on priority & status
Make an annoucement


Topic
Query
Priority
Status
Python
Week 4 Q1
+2
Solved.
Software Eng.
Quiz 1 Syllabus
+1
Unsolved
Python
Factorial Program
+2
Solved.
Queries Arranged based on priority & status
Make an annoucement


Topic
Query
Priority
Status
Make an announcement
+1
Unsolved
Python
Select Topic
+2
Solved
+2
Solved
The Python quiz is scheduled for 26th February 2023 from 2-6 pm IST
Syllabus : Week 1-4
Post
BACK

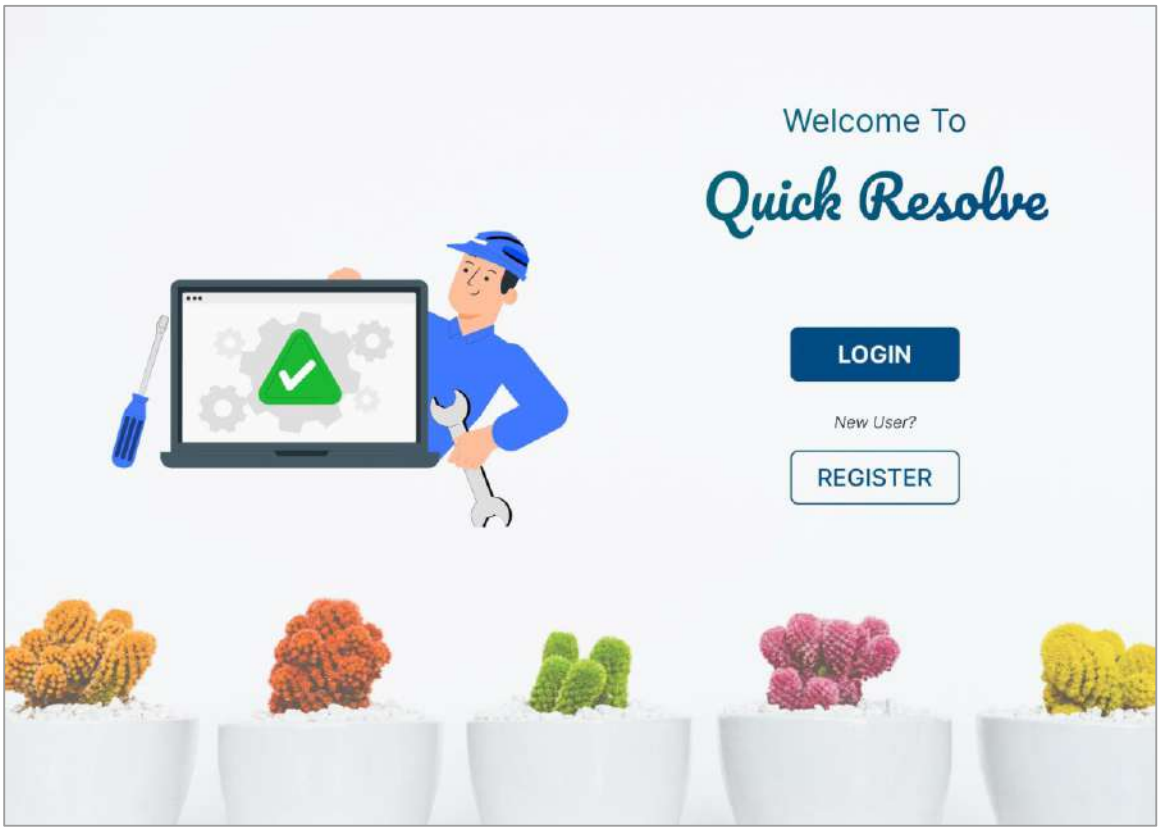
Component Design

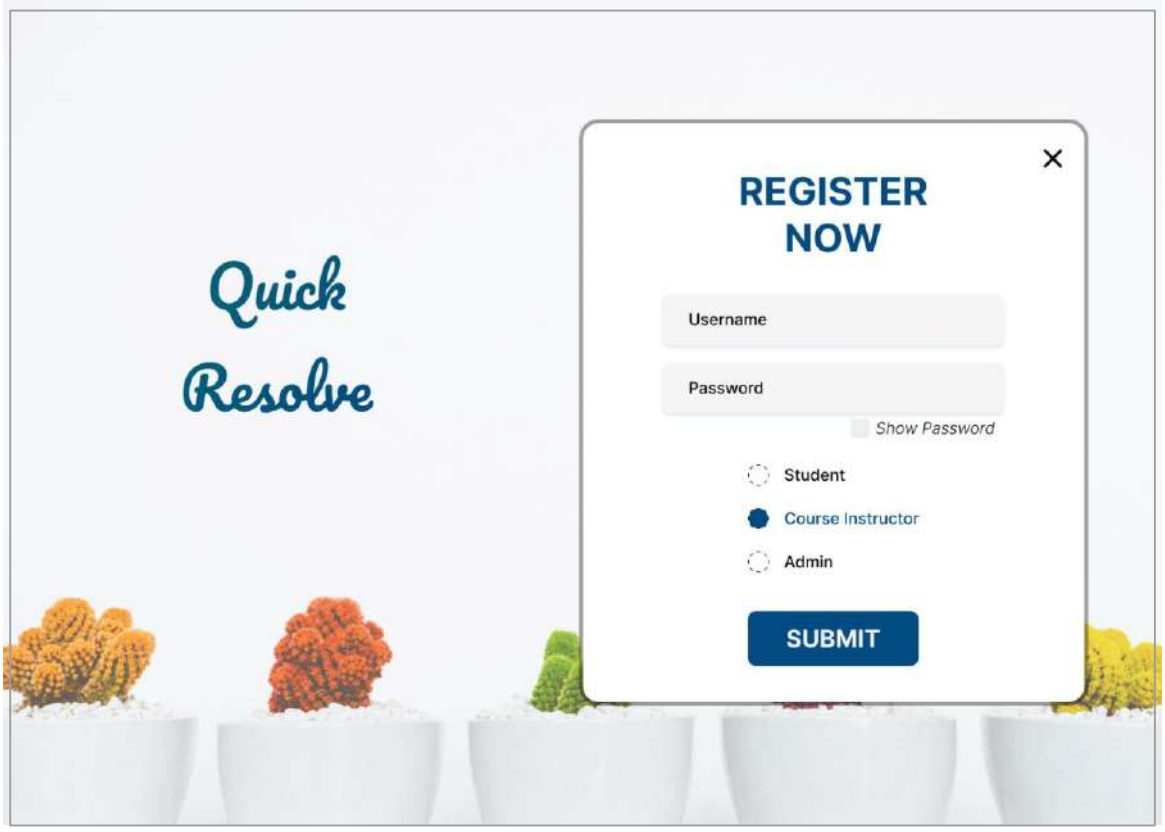
Design of components:

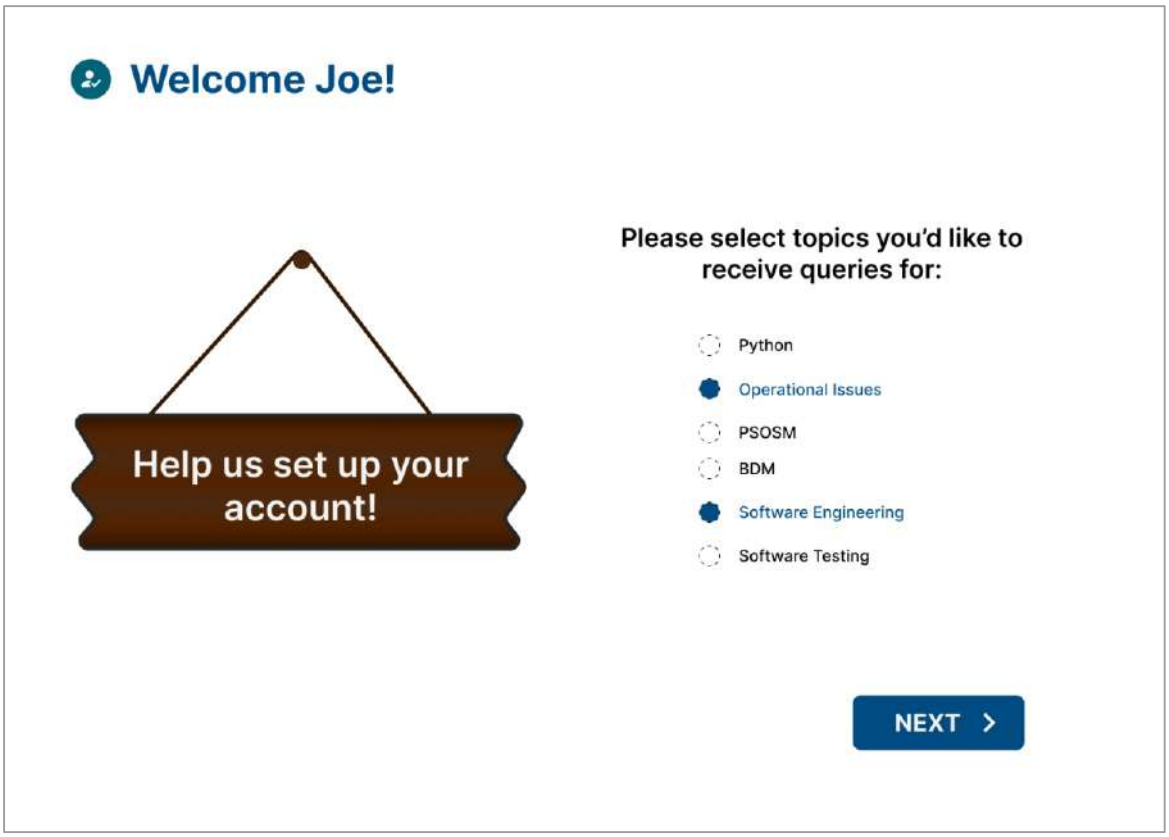
https://www.figma.com/file/0C0R7R7ohF8fR2Jadppvl1/Quick-Resolve?node-id=63%3A278&t=TKDEC5o9iWYueO1P-1

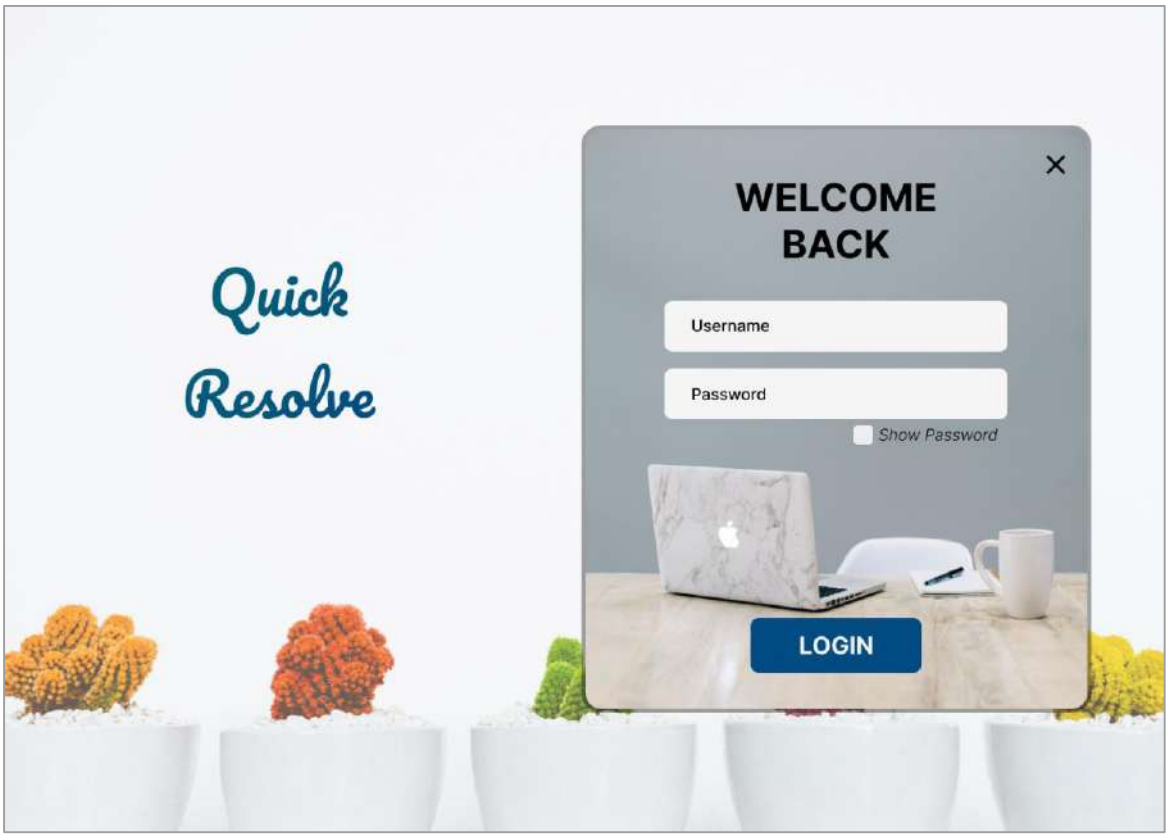
Prototype view:

https://www.figma.com/proto/0C0R7R7ohF8fR2Jadppvl1/Quick-Resolve?node-id=63%3A278&scaling=contain&page-id=0%3A1&starting-point-node-id=63%3A278


Welcome To
Quick Resolve
LOGIN
New User?
REGISTER


Quick
Resolve
REGISTER NOW
Username
Password
Show Password
Student
Course Instructor
Admin
SUBMIT


Welcome Joe!
Help us set up your account!
Please select topics you'd like to receive queries for:
Python
Operational Issues
PSOSM
BDM
Software Engineering
Software Testing
NEXT


Quick Resolve
WELCOME BACK
Username
Password
Show Password
LOGIN

Joe

Topic	Query	Priority	Status
Python	Week 4 Q1	+2	Unsolved
Soft. Engg	Quiz 1 Syllabus	+1	Solved
Python	Factorial program	+2	Solved

Joe

Factorial program — Author: Jay Pritchett

Sir can you please explain the recursive algo for computing the factorial?

Published on: 20/02/23 8:30 pm IST — +2

Joe ~ Course Instructor — Solved on: 20/02/23 9:57 pm IST

Herein, the recursive function will call itself until the value isn't equal to zero. The following formula is used:

BACK — PIN QUERY

Joe

Week 4 Q1 Author: Phil Dunphy

Sir could you please explain how the solution was derived?

Published on: 21/02/23 5:35 am IST +2

This query hasn't been solved yet!

Add a Solution

BACK PIN QUERY

Joe

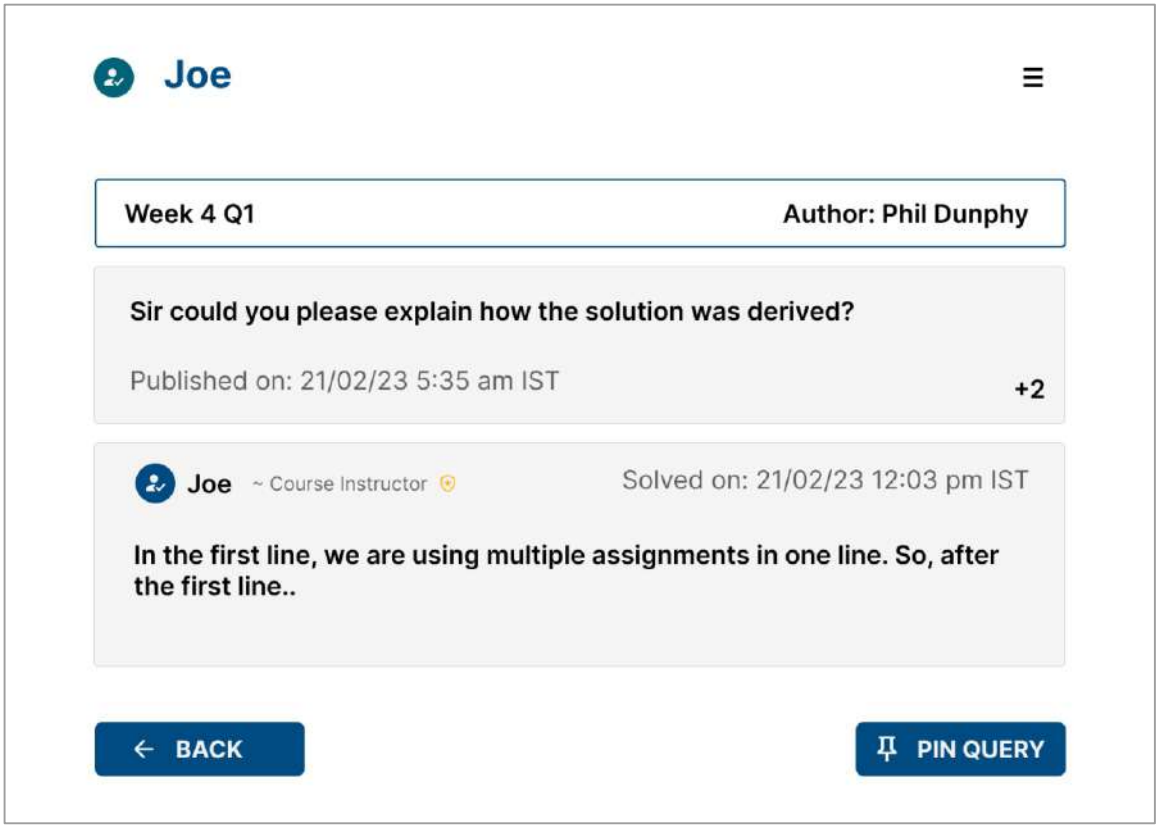
Week 4 Q1 Author: Phil Dunphy

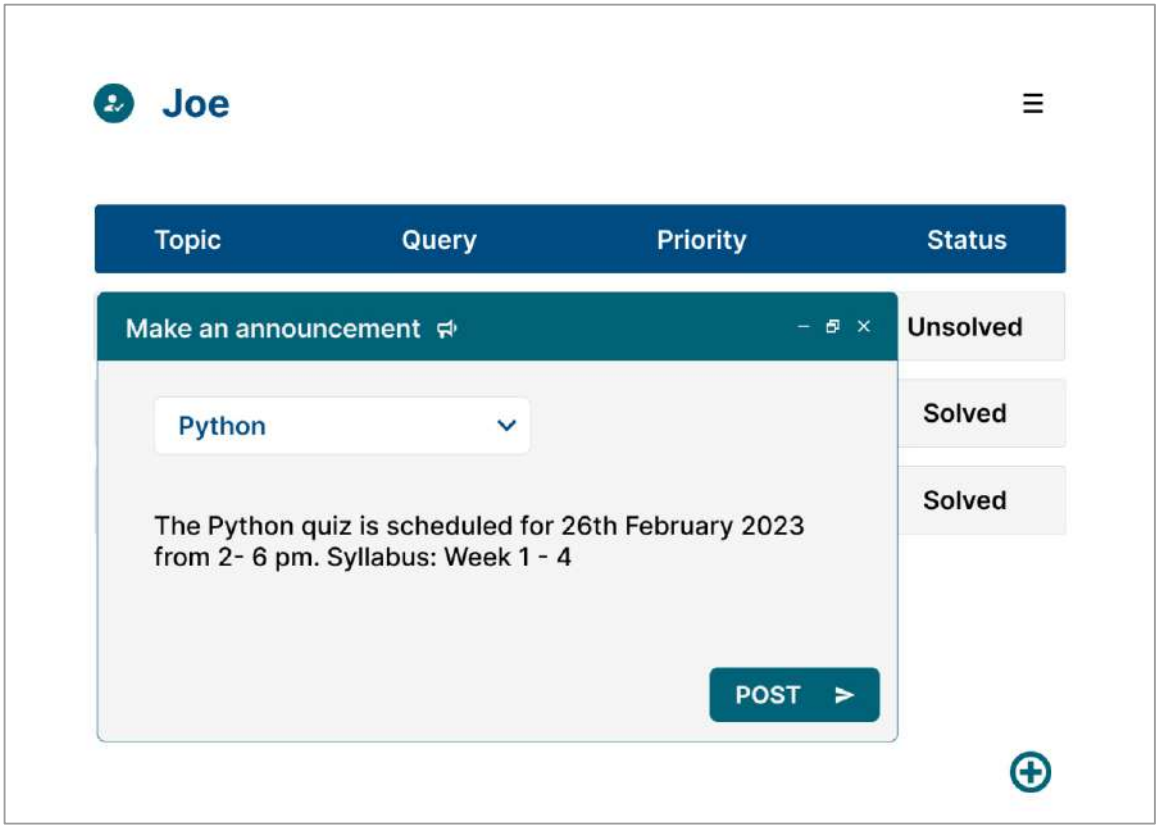
In the first line, we are using multiple assignments in one line. So, after the first line of execution, xa, y = b and z = c. In the second line, we Know that operator has the right to left associativity. So, the value of z which is c, will be assigned to y and the value of y which is now c, will be assigned to x. So finally, all variables will contain the same value c.

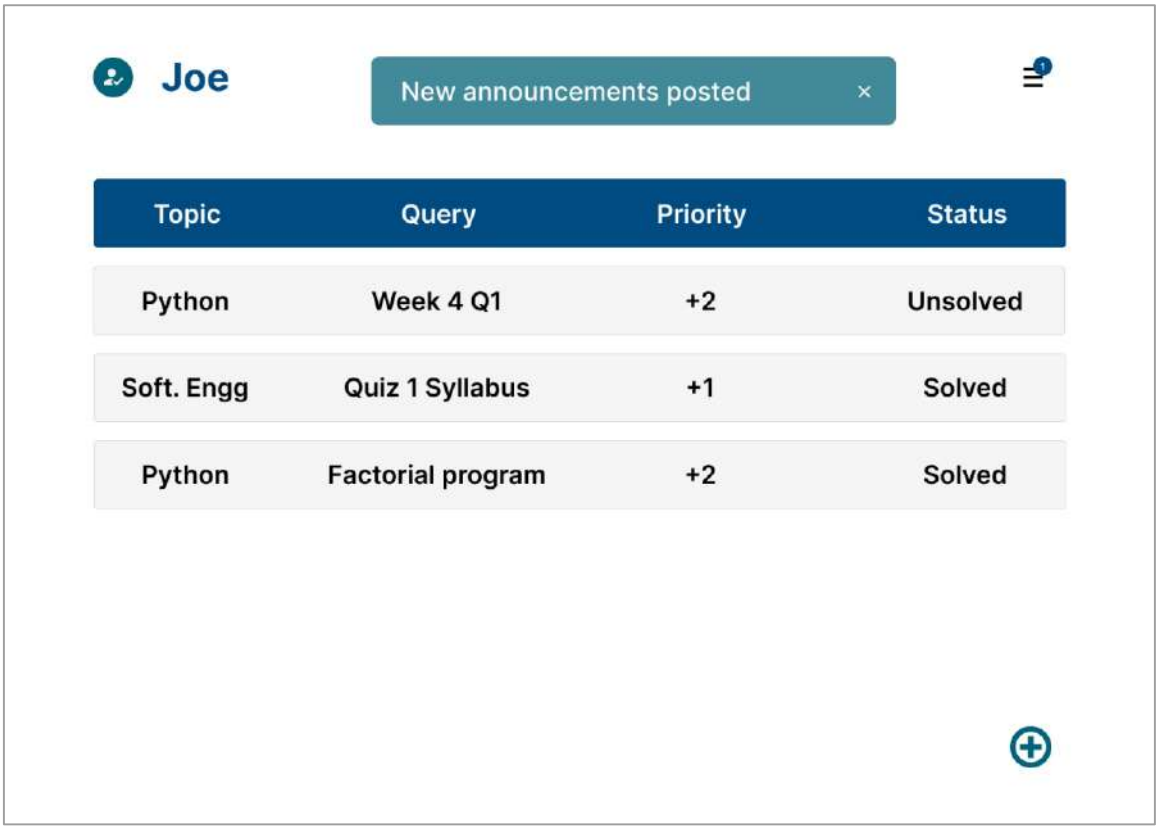
Hence, (a) and (d) are correct.

POST

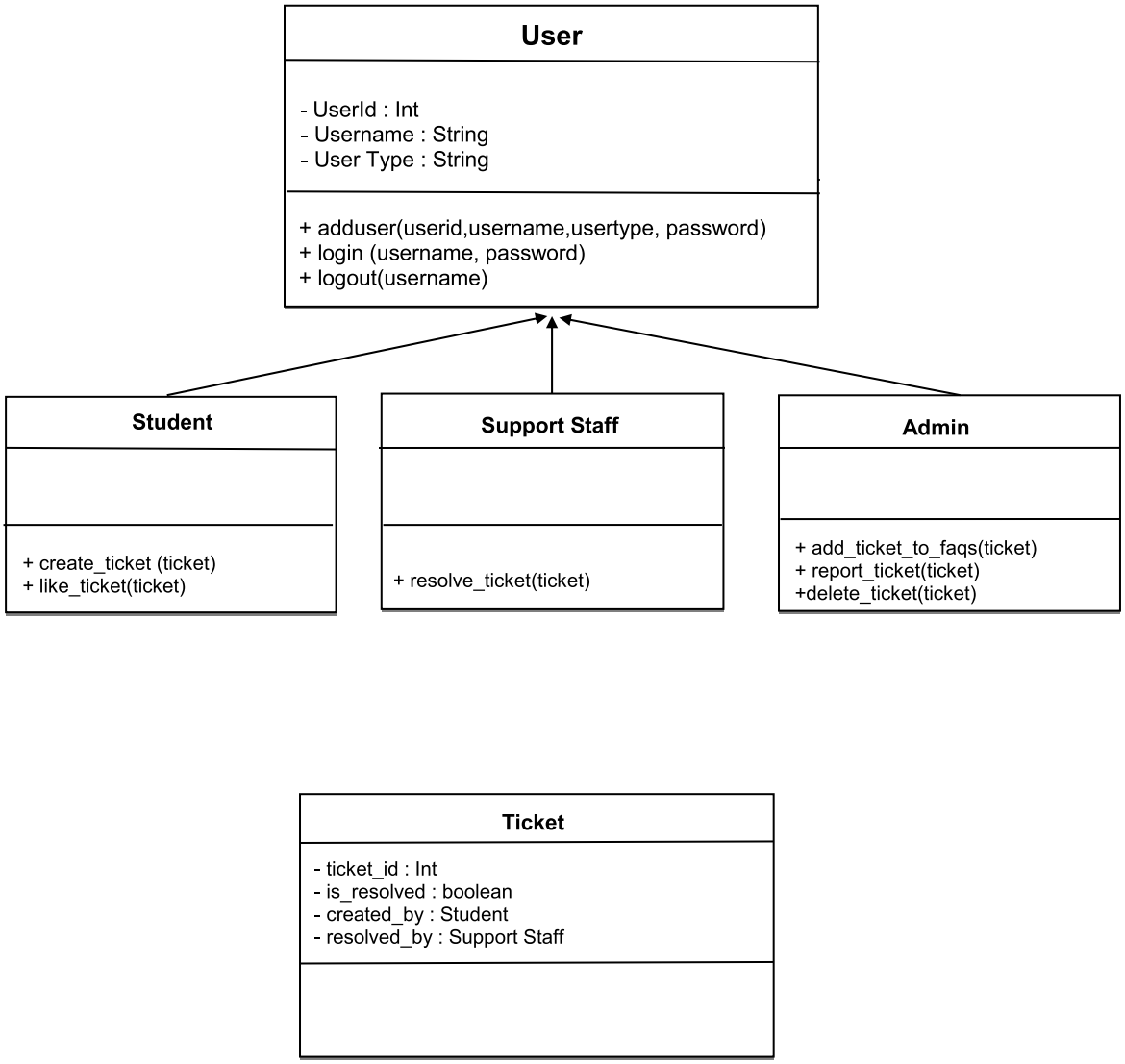
BACK PIN QUERY


Joe
Week 4 Q1
Author: Phil Dunphy
Sir could you please explain how the solution was derived?
Published on: 21/02/23 5:35 am IST
+2
Joe
~ Course Instructor
Solved on: 21/02/23 12:03 pm IST
In the first line, we are using multiple assignments in one line. So, after the first line..
BACK
PIN QUERY


Joe
Topic
Query
Priority
Status
Make an announcement
Unsolved
Python
Solved
Solved
The Python quiz is scheduled for 26th February 2023 from 2- 6 pm. Syllabus: Week 1 - 4
POST


Joe
New announcements posted
Topic
Query
Priority
Status
Python
Week 4 Q1
+2
Unsolved
Soft. Engg
Quiz 1 Syllabus
+1
Solved
Python
Factorial program
+2
Solved

Class Diagram



API Endpoint Decription

An API is an interface that two computer systems use to exchange information securely over the internet.

API documentation is essentially an instruction manual that explains how to use an API and its services. It is the primary resource for explaining what is possible with your API and how to get started.

The document should include examples of each call, parameter, and response to each call. Developers are the primary consumers of APIs and need comprehensive, accurate and well-structured information to start, thus the need for such a document.

For the purpose of this project, Swagger has been used to document the API.

File ▾ Edit ▾ Insert ▾ Generate Server ▾ Generate Client ▾ About ▾ Try our new Editor ↗

QuickResolve 1.0 OAS3

QuickResolve is a platform that enables effective query redressal. Herein, the students can post their queries and the support staff can resolve the queries. The platform provides additional functionalities such as pinning the queries, reporting, deleting & adding the queries to the FAQ document.The web application is built using the Flask framework & can be easily run on the localhost.

Here we present the API documentation for the designed application.

Servers

http://127.0.0.1:5000

User Endpoints for defining user operations

POST `/user/adduser/{userid}/{username}/{usertype}/{password}`

Add user to application

Parameters Try it out

Name	Description
userid * required `integer` *(path)*	179
username * required `string` *(path)*	Joey

Name	Description
usertype * required string *(path)*	Student
password * required string *(path)*	2pizzas@JoeySpecial

Request body **application/json**

Example Value Schema

```
{
  "UserId": 1,
  "Username": "John Doe",
  "UserType": "Support Staff",
  "Password": "1234@xyz"
}
```

Responses

Code	Description	Links
200	Request Successful - User Added Successfully! Media type **application/json** Controls `Accept` header. **Example Value** Schema `{ "userid": 179, "username": "Joey", "status": "User Added Successfully!" }`	*No links*
403	Already Exists Media type **application/json** **Example Value** Schema	*No links*

Code	Description	Links
	`{ "status": "User already exists" }`	
500	Internal Server Error Media type application/json **Example Value** Schema `{ "status": "Internal Server Error" }`	*No links*

GET `/user/login/{username}/{password}`

Operation to help user login to access resource.

Parameters Try it out

Name	Description
username * required `string` *(path)*	Joey
password * required `string` *(path)*	2pizzas@JoeySpecial

Responses

Code	Description	Links
200	Request Successful - User Verrified! Media type	*No links*

Code	Description	Links

	application/json
	Controls Accept header.
	Example Value Schema
	{
	"userid": 179,
	"username": "Joey",
	"usertype": "Student",
	"status": "User Verrified Successfully!"
	}	

404	Bad Request
	Media type
	application/json
	Example Value Schema
	{
	"status": "User does not exist"
	}
		No links

500	Internal Server Error
	Media type
	application/json
	Example Value Schema
	{
	"status": "Internal Server Error"
	}
		No links

DELETE /user/deluser/{username}/{password}

Operation to help users delete their account after validating credentials.

Parameters Try it out

Name	Description
username * required string *(path)*	Joey

Name	Description
password * required `string` *(path)*	2pizzas@JoeySpecial

Responses

Code	Description	Links
200	Request Successful - Account Deleted! Media type application/json Controls `Accept` header. **Example Value** Schema `{ "status": "Account Deleted Successfully!" }`	*No links*
404	Bad Request Media type application/json **Example Value** Schema `{ "status": "Account does not exist" }`	*No links*
500	Internal Server Error Media type application/json **Example Value** Schema `{ "status": "Internal Server Error" }`	*No links*

Support_Staff Endpoints for defining support staff operations

POST /support_staff/announcement/{userid}/{category}/{query}

Operation to help make announcements to the class

Parameters Try it out

Name	Description
userid * required integer *(path)*	179
category * required string *(path)*	Python
query * required string *(path)*	There is an error in the GA 4 Q7.. Could you please look into

Request body application/json

Example Value Schema


```
{
  "Ticket_id": "Python/12347",
  "Created_by": 179,
  "Solved_by": -1,
  "Category": "Python",
  "Query": "Dear All, the live session is scheduled for 20th March, 2023 at 17:30 pm. Kindly j
oin and make the most of teh oppurtunity.",
  "Created_at": "19th March 2023, 14:30 pm",
  "Solved_at": "-1",
  "IsResolved": true,
  "AddFAQ": false,
  "Pin": true,
  "Report": false,
  "Likes": 0
}
```


Responses

Code	Description	Links

200	Request Successful - Ticket Added Successfully!
	Media type
	application/json
	Controls Accept header.
	Example Value Schema
	{
	"userid": 179,
	"Ticket_id": "Joey",
	"status": "Ticket Added Successfully!"
	}
		No links

403	Already exists
	Media type
	application/json
	Example Value Schema
	{
	"status": "Ticket with same query already exists"
	}
		No links

500	Internal Server Error
	Media type
	application/json
	Example Value Schema
	{
	"status": "Internal Server Error"
	}
		No links

PUT `/support_staff/resolve_ticket/{ticket_id}`

Operation to help support staff resolve queries.

Parameters Try it out

Name	Description
ticket_id * required `string` *(path)*	Python_12345

Request body **application/json**

Example Value Schema

```
{
  "Ticket_id": "Python_12345",
  "IsResolved": true,
  "Solution": "Dear Student, GA 4 Q7 has been updated on the portal, kindly check.",
  "Solved_by": 143,
  "Solved_at": "19th March 2023, 19:47 pm"
}
```

Responses

Code	Description	Links
200	Request Successful - Query Resolved! Media type **application/json** Controls `Accept` header. **Example Value** Schema `{ "Ticket_id": "Python_12345", "status": "Query Resolved Successfully!" }`	*No links*
404	Bad Request Media type **application/json** **Example Value** Schema `{ "status": "Query with given id doesn't exist!" }`	*No links*

Code	Description	Links

500	Internal Server Error
	
	Media type
	application/json
	
	Example Value Schema
	
	{
	"status": "Internal Server Error"
	}
		No links

PUT `/support_staff/pin_ticket/{ticket_id}`

Operation to help pin important tickets.

Parameters Try it out

Name	Description
ticket_id * required `string` *(path)*	Python_12345

Request body application/json

Example Value Schema

```
{
  "Ticket_id": "Python_12345",
  "Pin": true
}
```

Responses

Code	Description	Links
200	Request Successful - Query Pinned!	*No links*

Code	Description	Links

	Media type
	application/json
	Controls Accept header.
	Example Value Schema
	{
	"Ticket_id": "Python_12345",
	"status": "Query Pinned successfully!"
	}	

404	Bad Request
	Media type
	application/json
	Example Value Schema
	{
	"status": "Query with given id doesn't exist!"
	}
		No links

500	Internal Server Error
	Media type
	application/json
	Example Value Schema
	{
	"status": "Internal Server Error"
	}
		No links

PUT /support_staff/report_ticket/{ticket_id}

Operation to help maintain decorum and policy defined by the university.

Parameters Try it out

Name	Description
ticket_id * required string *(path)*	Python_12345

Request body

application/json

Example Value Schema

```
{
  "Ticket_id": "Python_12345",
  "Report": true
}
```

Responses

Code	Description	Links
200	Request Successful - Query Reported! Media type application/json Controls Accept header. **Example Value** Schema `{ "Ticket_id": "Python_12345", "status": "Query Reported successfully!" }`	*No links*
404	Bad Request Media type application/json **Example Value** Schema `{ "status": "Query with given id doesn't exist!" }`	*No links*

Code	Description	Links

500	Internal Server Error
	
	Media type
	application/json
	
	Example Value Schema
	
	{
	"status": "Internal Server Error"
	}
		No links

Admin Endpoints for defining admin operations

PUT `/admin/add_ticket_to_faq/{ticket_id}`

Operation to help support staff resolve queries.

Parameters Try it out

Name	Description
ticket_id * required *string* *(path)*	Python_12345

Request body application/json

Example Value Schema

```
{
  "Ticket_id": "Python_12345",
  "AddFAQ": true
}
```

Responses

Code	Description	Links

200	Request Successful - Query Resolved!
	
	Media type
	application/json
	Controls Accept header.
	
	Example Value Schema
	
	{
	"Ticket_id": "Python_12345",
	"status": "Query Added to FAQ Successfully!"
	}
		No links

404	Bad Request
	
	Media type
	application/json
	
	Example Value Schema
	
	{
	"status": "Query with given id doesn't exist!"
	}
		No links

500	Internal Server Error
	
	Media type
	application/json
	
	Example Value Schema
	
	{
	"status": "Internal Server Error"
	}
		No links

DELETE /admin/delete_ticket/{ticket_id}

Operation to help delete redundant / objectionable tickets.

Parameters Try it out

Name	Description
ticket_id * required `string` *(path)*	Python_12345

Responses

Code	Description	Links

200	Request Successful - Query Deleted!
	
	Media type
	application/json
	Controls Accept header.
	
	Example Value Schema
	
	{
	"Ticket_id": "Python_12345",
	"status": "Query Deleted successfully!"
	}
		No links

404	Bad Request
	
	Media type
	application/json
	
	Example Value Schema
	
	{
	"status": "Query with given id doesn't exist!"
	}
		No links

500	Internal Server Error
	
	Media type
	application/json
	
	Example Value Schema
	
	{
	"status": "Internal Server Error"
	}
		No links

Student Endpoints for defining student operations

POST /Student/create_ticket/{userid}/{category}/{query}

Operation to help create tickets for student queries

Parameters Tryit out

Name	Description
userid * required integer *(path)*	179
category * required string *(path)*	Python
query * required string *(path)*	There is an error in the GA 4 Q7.. Could you please look into

Request body application/json

Example Value Schema


```
{
  "Ticket_id": "Python/12345",
  "Created_by": 179,
  "Solved_by": 0,
  "Category": "Python",
  "Query": "There is an error in the GA 4 Q7.. Could you please look into it?",
  "Created_at": "19th March 2023, 14:30 pm",
  "Solved_at": "19th March 2023, 19:47 pm",
  "IsResolved": false,
  "AddFAQ": false,
  "Pin": false,
  "Report": false,
  "Likes": 0
}
```

Responses

Code	Description	Links

200	Request Successful - Ticket Added Successfully!
	Media type
	application/json
	Controls Accept header.
	Example Value Schema
	{
	"userid": 179,
	"Ticket_id": "Joey",
	"status": "Ticket Added Successfully!"
	}
		No links

403	Already exists
	Media type
	application/json
	Example Value Schema
	{
	"status": "Ticket with same query already exists"
	}
		No links

500	Internal Server Error
	Media type
	application/json
	Example Value Schema
	{
	"status": "Internal Server Error"
	}
		No links

PUT /Student/like_ticket/{ticket_id}

Operation to help students like tickets of interest

Parameters Try it out

Name	Description
ticket_id * required `string` *(path)*	Python/12345

Request body **application/json**

Example Value Schema

```
{
  "Ticket_id": "Python/12345",
  "Likes": 1
}
```

Responses

Code	Description	Links
200	Request Successful - Ticket Liked Successfully! Media type **application/json** Controls Accept header. **Example Value** Schema `{ "userid": 179, "ticket_id": "Python/12345", "status": "Ticket Liked Successfully!" }`	*No links*
404	Bad Request Media type **application/json** **Example Value** Schema `{ "status": "Query with given id doesn't exist!" }`	*No links*
500	Internal Server Error	*No links*

Code	Description	Links

Media type

application/json

Example Value Schema

```
{
  "status": "Internal Server Error"
}
```

FAQ Endpoints for defining FAQ operations

POST /FAQ/add_ticket/{ticket_id}

Operation to help add popular queries to the FAQ document.

Parameters Try it out

Name	Description
ticket_id * required string *(path)*	Python/12345

Request body application/json

Example Value Schema

```
{
  "Ticket_id": "Python/12345",
  "Query": "What is the quiz-1 syllabus?",
  "Solution": "The Quiz-1 syllabus is Week 1-4."
}
```

Responses

Code	Description	Links

200	Request Successful - Ticket Added to FAQ Successfully!
	
	Media type
	application/json
	Controls Accept header.
	
	Example Value Schema
	
	{
	"Q_Id": 34,
	"status": "FAQ doc updated Successfully!"
	}
		No links

403	Already Exists
	
	Media type
	application/json
	
	Example Value Schema
	
	{
	"status": "Query already exists in FAQs!"
	}
		No links

500	Internal Server Error
	
	Media type
	application/json
	
	Example Value Schema
	
	{
	"status": "Internal Server Error"
	}
		No links

Dashboard Endpoints for defining dashboard operations

GET /dashboard/tickets/{category}

Operation to help users access all the tickets created in the respective categories.

Parameters Try it out

Name	Description
category * required `string` *(path)*	Python

Responses

Code	Description	Links
200	Request Successful - Tickets Returned! Media type **application/json** Controls `Accept` header. **Example Value** Schema `{ "Ticket_id": "Python/12345", "Created_by": 179, "Solved_by": 0, "Category": "Python", "Query": "There is an error in the GA 4 Q7.. Could you please look into it?", "Created_at": "19th March 2023, 14:30 pm", "Solved_at": "19th March 2023, 19:47 pm", "IsResolved": false, "AddFAQ": false, "Pin": false, "Report": false, "Likes": 0 }`	*No links*
404	Bad Request Media type **application/json** **Example Value** Schema `{ "status": "Tickets do not exist for the given category" }`	*No links*
500	Internal Server Error	*No links*

Code	Description	Links

Media type

application/json

Example Value Schema

```
{
  "status": "Internal Server Error"
}
```

Schemas

user {

```
UserId      [...]
UserName    [...]
UserType    [...]
Password    [...]
}
```

faq {

```
Q_Id        [...]
Query       [...]
Solution    [...]
Category    [...]
}
```

ticket {

```
    Ticket_id        [...]
    Created_by       [...]
    Solved_by        [...]
    Category         [...]
    Query            [...]
    Solution         [...]
    Created_at       [...]
    Solved_at        [...]
    IsResolved       [...]
    AddFAQ           [...]
    Pin              [...]
    Report           [...]
    Likes            [...]
}
```

Test Driven Development (TDD)

Test Driven Development is used especially in Agile Processes.

Herein, we first write the tests for the functionality that we wish to implement. Thereafter we design and develop that functionality.

The first thing we do is express the requirement / feature / user story in the form of a test and then we run this created test. And since we haven't yet implemented the feature in question, the test will fail. So we deliberately create a test and make it fail. Next, our goal is to somehow make this test pass & to accomplish this, we create the minimum code to meet the needs of the test.

Finally we refactor the code, i.e. we try to improve the quality, make the code more modular and elegant while still satisfying the test case scenarios defined.

Thus the mantra for Test Driven Development can be summarized as:

- **RED:** Write a Test and See it Fail
- **GREEN:** Write code and See it Pass
- **REFACTOR:** Make the code better

Test Cases

Note: **The test cases have been designed in the following format:**

[Page Being tested,
Inputs,
Expected Output,
Actual Output,
Result – Success/Fail]

Dashboard – Load Tickets API:

- ## Testing for server error when the database has not been created
  [User Dashboard,
  Category – “Python”,
  {“message”: “Internal Server Error”},
  {“message”: “Internal Server Error”},
  Success]
- ## testing for tickets when table not creating
  [User Dashboard,
  Category – “Python”,
  {“message”: “Tickets do not exist for the given category”},
  {“message”: “Internal Server Error”},
  Fail]
- ## Testing for tickets when empty table is created
  [User Dashboard,
  Category – “Python”,
  {“message”: “Tickets do not exist for the given category”},
  {“message”: “Tickets do not exist for the given category”},
  Success]
- ## Testing for tickets when table has data but query hasn’t been defined
  [User Dashboard,
  Category – “Python”,
  [{'Ticket_id': 'Python/1', 'Created_by': 1, 'Solved_by': 2, 'Category': 'Python', 'Query': 'There is an error in the GA 4 Q7.. Could you please look into it?', 'Solution': 'Dear Student, the quetsion has been updated. Please check.', 'Created_at': '03/23/2023 @ 4:31am', 'Solved_at': '03/23/2023 @ 10:11am', 'IsResolved': 'True', 'AddFAQ': 'False', 'Pin': 'False', 'Report': 'False', 'Likes': 3}],
  {“message”: “Internal Server Error”},
  Fail]

- ## Testing for tickets when table has data & query has been defined
  [User Dashboard,
  Category – “Python”,
  [{'Ticket_id': 'Python/1', 'Created_by': 1, 'Solved_by': 2, 'Category': 'Python', 'Query': 'There is an error in the GA 4 Q7.. Could you please look into it?', 'Solution': 'Dear Student, the quetsion has been updated. Please check.', 'Created_at': '03/23/2023 @ 4:31am', 'Solved_at': '03/23/2023 @ 10:11am', 'IsResolved': 'True', 'AddFAQ': 'False', 'Pin': 'False', 'Report': 'False', 'Likes': 3}],
  [{'Ticket_id': 'Python/1', 'Created_by': 1, 'Solved_by': 2, 'Category': 'Python', 'Query': 'There is an error in the GA 4 Q7.. Could you please look into it?', 'Solution': 'Dear Student, the quetsion has been updated. Please check.', 'Created_at': '03/23/2023 @ 4:31am', 'Solved_at': '03/23/2023 @ 10:11am', 'IsResolved': 'True', 'AddFAQ': 'False', 'Pin': 'False', 'Report': 'False', 'Likes': 3}],
  Success]

As a result of creating & analyzing the test cases, the following API endpoint was designed to help display the tickets created in the various categories on the user dashboard:

```
class DashboardApi(Resource):

  def get(self, cat):
    dashboard = db.session.execute(f"SELECT * FROM ticket WHERE Category = '{cat}'").fetchall()
    dash = list(dashboard)
    dashboard = [dict(row) for row in dashboard]

    if dash == [] :
      abort(404, message="Tickets do not exist for the given category")

    return dashboard

api.add_resource(DashboardApi, "/dashboard/tickets/<cat>")
```

Admin - Select Tickets for FAQ API:

- ## Testing for server error when the ticket with said id does not exist
  [FAQ Document,
  Ticket_id – “Python91”,
  {“message”: “Query with given id doesn't exist”},
  {“message”: “Internal Server Error”},
  Fail]
- ## Testing for server error when the ticket with said id does not exist after defining query
  [FAQ Document,

Ticket_id – “Python91”,
{“message”: “Query with given id doesn't exist”},
{“message”: “Query with given id doesn't exist”},
Success]

- ## Testing for server error when the query with said id does exist
  [FAQ Document,
  Ticket_id – “Python1”,
  {'Ticket_id': 'Python1', 'status': 'Query AddFAQ status updated!'},
  {“message”: “Internal Server Error”},
  Fail]
- ## Testing for server error when the ticket with said id does exist after defining the query
  [FAQ Document,
  Ticket_id – “Python1”,
  {'Ticket_id': 'Python1', 'status': 'Query AddFAQ status updated!'},
  {'Ticket_id': 'Python1', 'status': 'Query AddFAQ status updated!'},
  Success]

As a result of creating & analyzing the test cases, the following API endpoint was designed to help display the admins select the tickets to be added to the FAQ document:

```
add_faq = reqparse.RequestParser(bundle_errors=True)
add_faq.add_argument("Ticket_id", type=str, required=True, help="No ticket id provided")

class AdminApi(Resource):

  def put(self):

    args = add_faq.parse_args()
    Ticket_id = args.get("Ticket_id")
    tic = db.session.execute(f"SELECT * FROM ticket WHERE Ticket_id = '{Ticket_id}'")
    tic = [dict(row) for row in tic]

    if tic == []:
      abort(404, message="Query with given id doesn't exist")

    tru = True
    db.session.execute(f"UPDATE ticket SET AddFAQ = '{tru}' WHERE Ticket_id = '{Ticket_id}'")
    db.session.commit()

    response = { "Ticket_id": Ticket_id, "status": "Query AddFAQ status updated!" }

    return response

api.add_resource(AdminApi, "/admin/add_ticket_to_faq")
```

Admin - Delete reported tickets API:

- ## Testing for server error when the ticket with said id does not exist
  [User Dashboard,
  Ticket_id – “Python91”,
  {“message”: “Query with given id doesn't exist”},
  {“message”: “Internal Server Error”},
  Fail]
- ## Testing for server error when the ticket with said id does not exist after defining query
  [User Dashboard,
  Ticket_id – “Python91”,
  {“message”: “Query with given id doesn't exist”},
  {“message”: “Query with given id doesn't exist”},
  Success]
- ## Testing for server error when the ticket with said id does exist
  [User Dashboard,
  Ticket_id – “English7”,
  {'Ticket_id': 'English7', 'status': 'Query Deleted Successfully!'},
  {“message”: “Internal Server Error”},
  Fail]
- ## Testing for server error when the ticket with said id does exist but is not reported after defining query
  [User Dashboard,
  Ticket_id – “English7”,
  {'message': "Operation not allowed as Query hasn't been reported yet."},
  {'Ticket_id': 'English7', 'status': 'Query Deleted Successfully!'},
  Fail]
- ## Testing for server error when the ticket with said id does exist but is not reported after defining logic
  [User Dashboard,
  Ticket_id – “English7”,
  {'message': "Operation not allowed as Query hasn't been reported yet."},
  {'message': "Operation not allowed as Query hasn't been reported yet."},
  Success]
- ## Testing for server error when the ticket with said id does exist and has been reported after defining query
  [User Dashboard,
  Ticket_id – “English7”,
  {'Ticket_id': 'English7', 'status': 'Query Deleted Successfully!'},
  {'Ticket_id': 'English7', 'status': 'Query Deleted Successfully!'},
  Success]

As a result of creating & analyzing the test cases, the following API endpoint was designed to help display the admins delete the tickets reported by the Support Staff:

```
del_ticket = reqparse.RequestParser(bundle_errors=True)
del_ticket.add_argument("Ticket_id", type=str, required=True, help="No ticket id provided")

class Delete_ticket(Resource):

  def delete(self):

    args = del_ticket.parse_args()
    Ticket_id = args.get("Ticket_id")
    tru = True
    tic = db.session.execute(f"SELECT * FROM ticket WHERE Ticket_id = '{Ticket_id}'")

    tic = [dict(row) for row in tic]

    if tic == []:
      abort(404, message="Query with given id doesn't exist")
    elif tic[0]['Report'] != 'True':
      abort(405, message="Operation not allowed as Query hasn't been reported yet.")

    db.session.execute(f"DELETE FROM ticket WHERE Ticket_id = '{Ticket_id}'")
    db.session.commit()

    response = { "Ticket_id": Ticket_id, "status": "Query Deleted Successfully!" }

    return response

api.add_resource(Delete_ticket, "/admin/delete_ticket")
```

FAQ – Adding Selected queries to doc API:

- ## Testing for server error when the ticket with said id already exists in the doc
  [FAQ Document,
  Ticket_id – “Python91”,
  {“message”: “Query already exists in FAQs!”},
  {“message”: “Internal Server Error”},
  Fail]
- ## Testing for server error when the ticket with said id already exists in the doc after defining logic
  [FAQ Document,
  Ticket_id – “Python91”,
  {“message”: “Query already exists in FAQs!”},
  {“message”: “Query already exists in FAQs!”},

Success]

- ## Testing for server error when the query with said id does not exist in the doc
  [FAQ Document,
  Ticket_id – “Python1”,
  {'Ticket_id': 'Python1', 'status': 'FAQ doc updated successfully!'},
  {“message”: “Internal Server Error”},
  Fail]
- ## Testing for server error when the ticket with said id does not exist in the doc after defining the query
  [FAQ Document,
  Ticket_id – “Python1”,
  {'Ticket_id': 'Python1', 'status': 'FAQ doc updated successfully!'},
  {'Ticket_id': 'Python1', 'status': 'FAQ doc updated successfully!'},
  Success]

As a result of creating & analyzing the test cases, the following API endpoint was designed to help add the queries to the FAQ document:

```
create_faq = reqparse.RequestParser(bundle_errors=True)
create_faq.add_argument("Ticket_id", type=str, required=True, help="No ticket id provided")
create_faq.add_argument("Query", type=str, required=True, help="No query provided")
create_faq.add_argument("Solution", type=str, required=True, help="No solution provided")
create_faq.add_argument("Category", type=str, required=True, help="No category provided")

class FAQApi(Resource):

  def post(self):

    args = create_faq.parse_args()
    Ticket_id = args.get("Ticket_id")
    Query = args.get("Query")
    Solution = args.get("Solution")
    Category = args.get("Category")
    tic = db.session.execute(f"SELECT * FROM faq WHERE Query = '{Query}'")
    tic = [dict(row) for row in tic]

    if tic != []:
      abort(409, message="Query already exists in FAQs!")

    query_faq = faq(Query = Query, Solution = Solution, Category = Category)
    db.session.add(query_faq)
    db.session.commit()

    Q_Id = query_faq.Q_Id
    response = { "Ticket_Id" : Ticket_id, "status" : "FAQ doc updated Successfully!" }

    return response
```

api.add_resource(FAQApi, "/FAQ/add_ticket")

```
Command Prompt
INFO     sqlalchemy.engine.Engine:log.py:117 COMMIT
INFO     sqlalchemy.engine.Engine:log.py:117 BEGIN (implicit)
INFO     sqlalchemy.engine.Engine:log.py:117 SELECT faq."Q_Id" AS "faq_Q_Id", faq."Query" AS "faq_Query", faq."Solution"
 AS "faq_Solution", faq."Category" AS "faq_Category"
FROM faq
WHERE faq."Q_Id" = ?
INFO     sqlalchemy.engine.Engine:log.py:117 [generated in 0.00059s] (1,)
=========================================== short test summary info ============================================
FAILED Code/test_main.py::test_delete_ticket - AssertionError: assert '405 METHOD NOT ALLOWED' == '200 OK'
FAILED Code/test_main.py::test_add_faq - AssertionError: assert {'Ticket_Id': 'Python1', 'status': 'FAQ doc updated Succ
essfully!'} == {'Ticket_Id': 1, 'sta...
========================================== 2 failed, 3 passed in 1.67s ========================================

C:\Users\lenovo\Desktop\IITM SE Project>pytest -vv
============================================= test session starts ==============================================
platform win32 -- Python 3.8.0, pytest-7.2.2, pluggy-1.0.0 -- c:\users\lenovo\appdata\local\programs\python\python38\pyt
hon.exe
cachedir: .pytest_cache
rootdir: C:\Users\lenovo\Desktop\IITM SE Project
collected 5 items

Code/test_main.py::test_dashboard_tickets PASSED                                                          [ 20%]
Code/test_main.py::test_faq_select PASSED                                                                 [ 40%]
Code/test_main.py::test_delete_ticket PASSED                                                              [ 60%]
Code/test_main.py::test_add_faq PASSED                                                                    [ 80%]
Code/test_main.py::test_user PASSED                                                                       [100%]

============================================= 5 passed in 4.41s ================================================

C:\Users\lenovo\Desktop\IITM SE Project>
```

Sign Up – Helping users access the software API:

[

- Sign up Api being Tested,
  Inputs : { “user_name” : “John Doe”, “User_password” : “12345”, “User_email” : “abc@xyz.com”, “User_type” : ‘Student” },
  Expected outputs : [{ Status_code : 201, Description : “ User added successfully “, }, { Status_code : 409, Description : “ user with same name or email already exist “ }],
  Actual Output : { Status_code : 201, Description : “ User added successfully “, },
  Success

]

[

- Sign up Page being Test,
  Inputs : { “user_name” : “John Doe”, “User_password” : “12345”, “User_email” : “abc@xyz.com” “User_type” : ‘Student” },
  Expected outputs : [{ Status_code : 201, Description : “ User added successfully “, }, { Status_code : 409, Description : “ user with same name or email already exist “ }],
  Actual Output : { Status_code : 409, Description : “user with same name or email already exist “, },
  Success

]

Resolve – Helping support team solve tickets API:

[

- Resolve Ticket being Tested,
  Inputs : { "ticket_id": 1, "ticket_query": "Whats is syallbus for quiz 2", "ticket_answer": "week 1-8”, “User_type” : “support staff” },
  Expected outputs : [{ Status_code : 200, Message : “ticket resolve” }, { Status_code : 400 Message : “Bad request ! Only a support staff can resolve a ticket” }],
  Actual Output : { Status_code : 200, Message : “ticket resolve” },
  Success

]

[

- Resolve Ticket being Tested,
  Inputs : { "ticket_id": 1, "ticket_query": "Whats is syallbus for quiz 2", "ticket_answer": "week 1-8”, “User_type” : “Student” },

Expected outputs : [{ Status_code : 200, Message : “ticket resolve” }, { Status_code : 400 Message : “Bad request ! Only a support staff can resolve a ticket” }],

Actual Output : { Status_code : 400 Message : “Bad request ! Only a support staff can resolve a ticket” },

Success

]

Create Ticket – Helping students create tickets API:

- ## Creating a valid user ticket
  [
  Create Ticket(create_ticket endpoint),
  {"Ticket_id": "T001", "Created_by": 1, "Solved_by": null, "Category": "Software", "Query": "How to install Python", "Created_at": "2022-03-23 10:00:00", "Solved_at": null, "IsResolved": false, "AddFAQ": false, "Pin": false, "Report": false, "Likes": 0},
  {"userid": 1, "ticket_id": "T001", "status": "Ticket Added Successfully!"},
  {"userid": 1, "ticket_id": "T001", "status": "Ticket Added Successfully!"}
  Success
  ]
- ## Creating an invalid user ticket
  [
  Create Ticket(create_ticket endpoint),
  {"Ticket_id": "T002", "Created_by": 1, "Solved_by": null, "Category": "Software", "Query": "How to install Python", "Created_at": "2022-03-23 10:00:00", "Solved_at": null, "IsResolved": false, "AddFAQ": false, "Pin": false, "Report": false, "Likes": 0},
  {"status": "Ticket with same query already exists"},
  {"status": "Ticket with same query already exists"},
  Fail
  ]

As a result of creating & analyzing the test cases, the following API endpoint was designed to help students create tickets & put their queries forth the support team:

```
@app.route('/Student/create_ticket///', methods=['POST'])
def create_ticket(userid, category, query):
        # Extract ticket details from request body
        data = request.get_json()
        ticket_id = data['Ticket_id']
        created_by = data['Created_by']
        solved_by = data['Solved_by']
        created_at = data['Created_at']
        solved_at = data['Solved_at']
        is_resolved = data['IsResolved']
        add_faq = data['AddFAQ']
        pin = data['Pin']
        report = data['Report']
        likes = data.get('Likes', 0)
        # Check if ticket with same query already exists
        if Ticket.query.filter_by(category=category, query=query).first():
                return jsonify({'status': 'Ticket with same query already exists'}), 403
        # Create new ticket
```

```
ticket = Ticket(
        ticket_id=ticket_id,
        created_by=created_by,
        solved_by=solved_by,
        category=category,
        query=query,
        created_at=created_at,
        solved_at=solved_at,
        is_resolved=is_resolved,
        add_faq=add_faq,
        pin=pin,
        report=report,
        likes=likes
)
db.session.add(ticket)
db.session.commit()

# Return success message
return jsonify({'userid': userid, 'ticket_id': ticket_id, 'status': 'Ticket Added Successfully!'}),
200
```

Like Ticket – Helping students +1 / like existing tickets API:

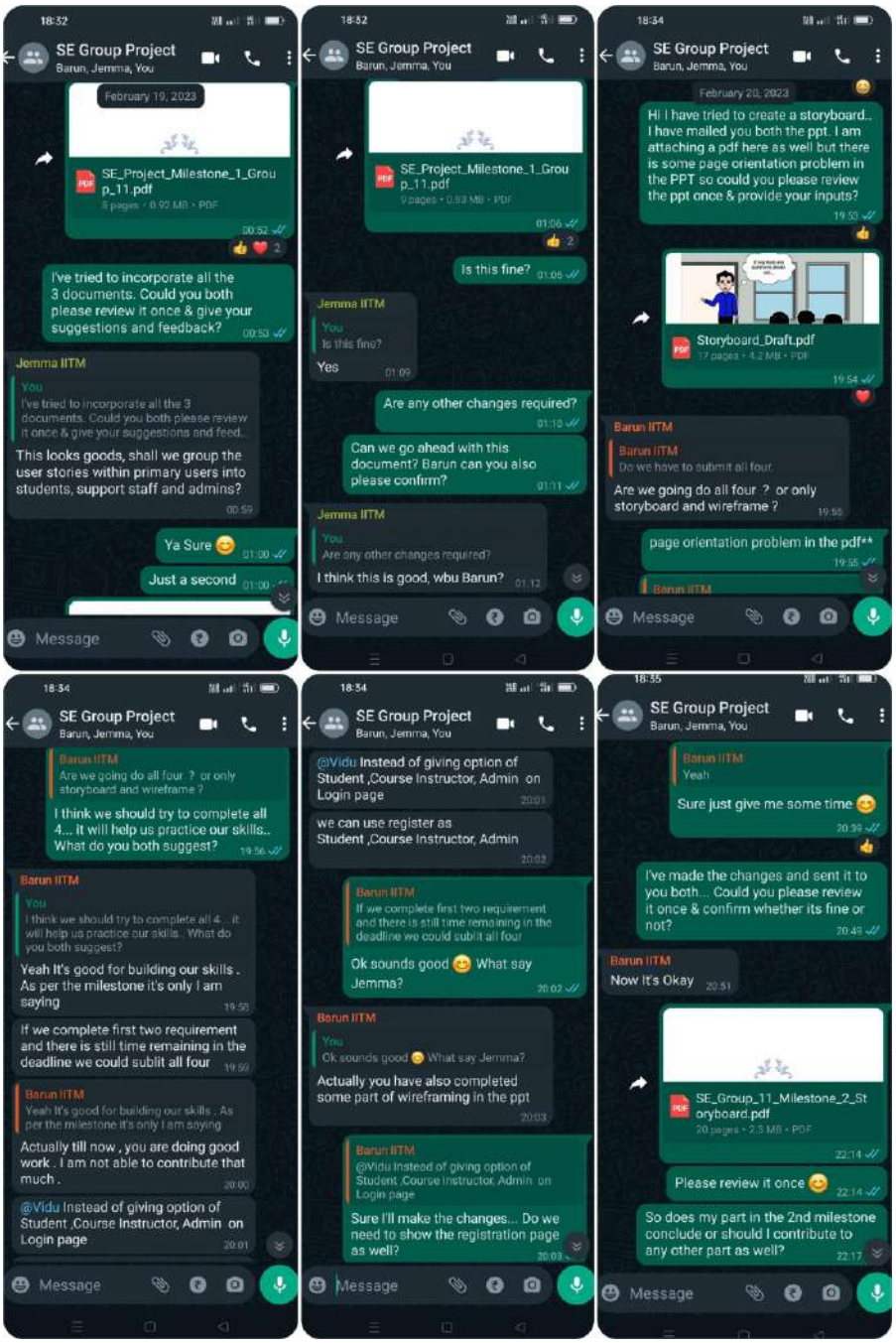
- ## Like a valid ticket
  [
  Like Ticket(like_ticket endpoint),
  {"Likes": 1},
  {"userid": 1, "ticket_id": "T001", "status": "Ticket Liked Successfully!"},
  {"userid": 1, "ticket_id": "T001", "status": "Ticket Liked Successfully!"},
  Success
  ]
- ## Like an invalid ticket
  [
  Like Ticket(like_ticket endpoint),
  {"Likes": 1},
  {"status": "Query with given id doesn't exist!"},
  {"status": "Query with given id doesn't exist!"},
  Fail
  ]

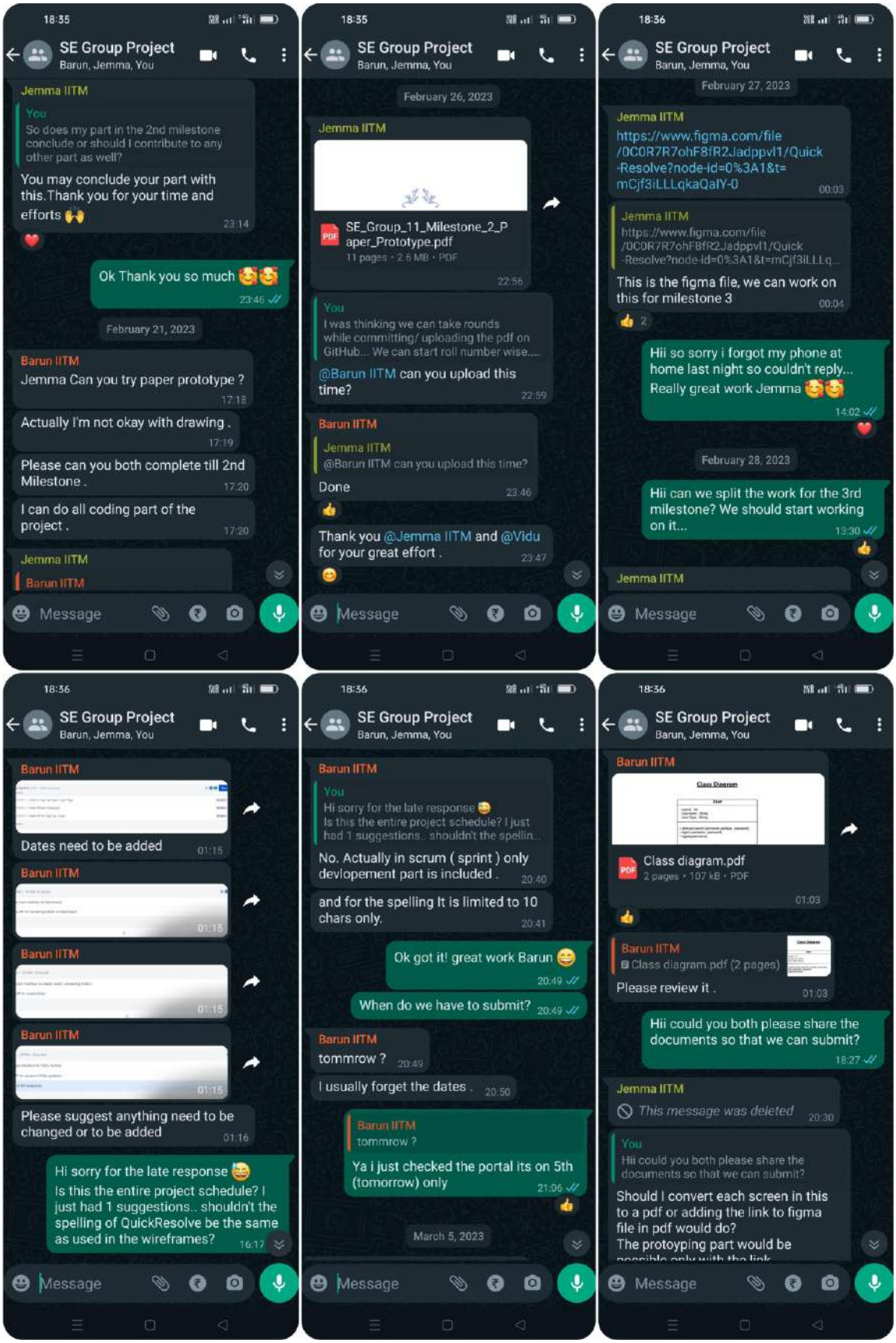
As a result of creating & analyzing the test cases, the following API endpoint was designed to help students +1 the existing queries to help prevent duplications:

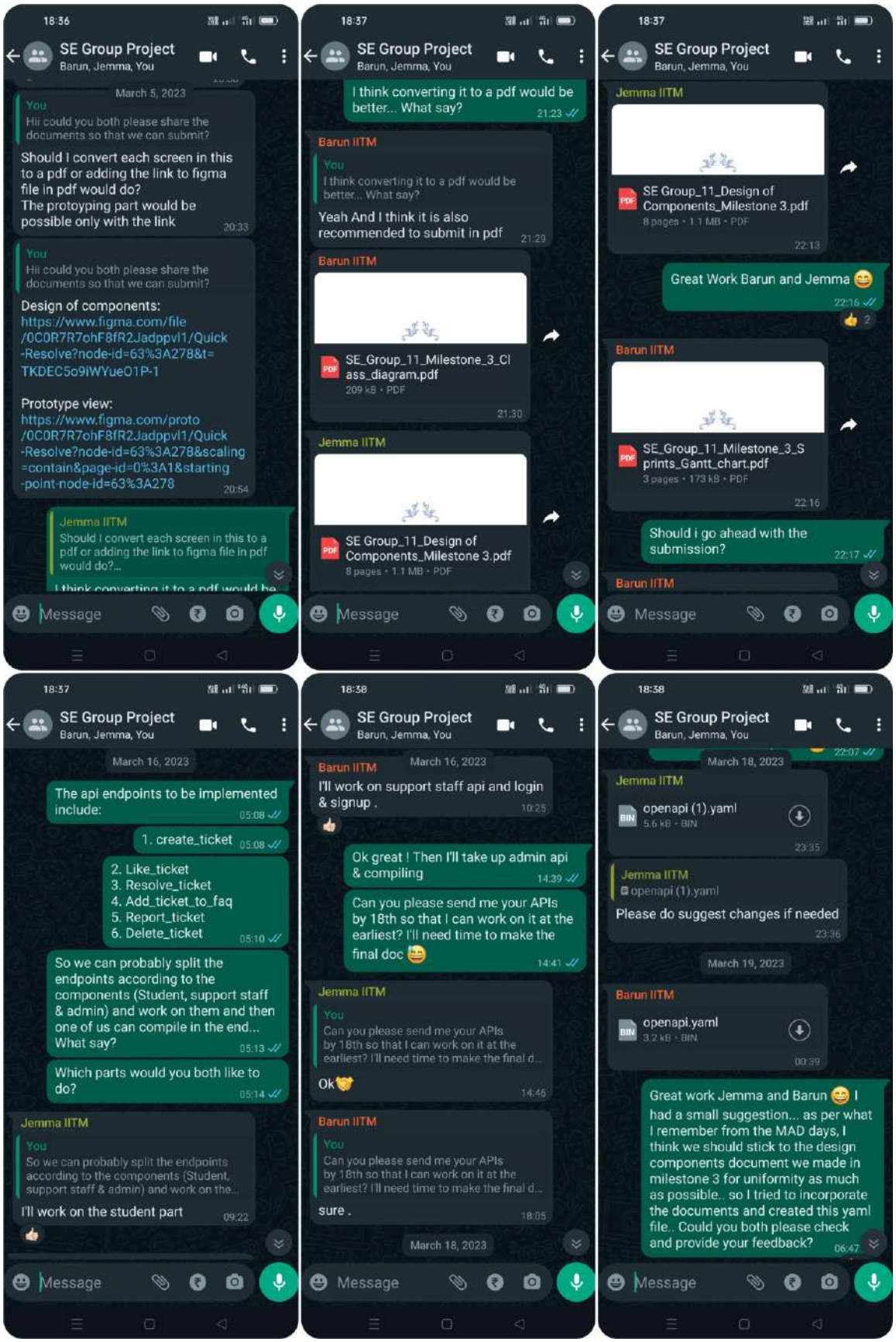
```
@app.route('/Student/like_ticket/', methods=['PUT'])
def like_ticket(ticket_id):
	# Get the request body
	request_body = request.get_json()
	# Get the likes from the request body
	likes = request_body.get('Likes')
	# Check if the ticket exists in the database
	ticket = Ticket.query.filter_by(ticket_id=ticket_id).first()
	if not ticket:
		return jsonify({'status': 'Query with given id doesn\'t exist!'}), 404
	# Update the likes of the ticket
	ticket.likes += likes

	# Commit the changes to the database
	db.session.commit()

	# Return a success message
	return jsonify({'userid': ticket.created_by, 'ticket_id': ticket.ticket_id, 'status': 'Ticket Liked
	Successfully!'}), 200
```

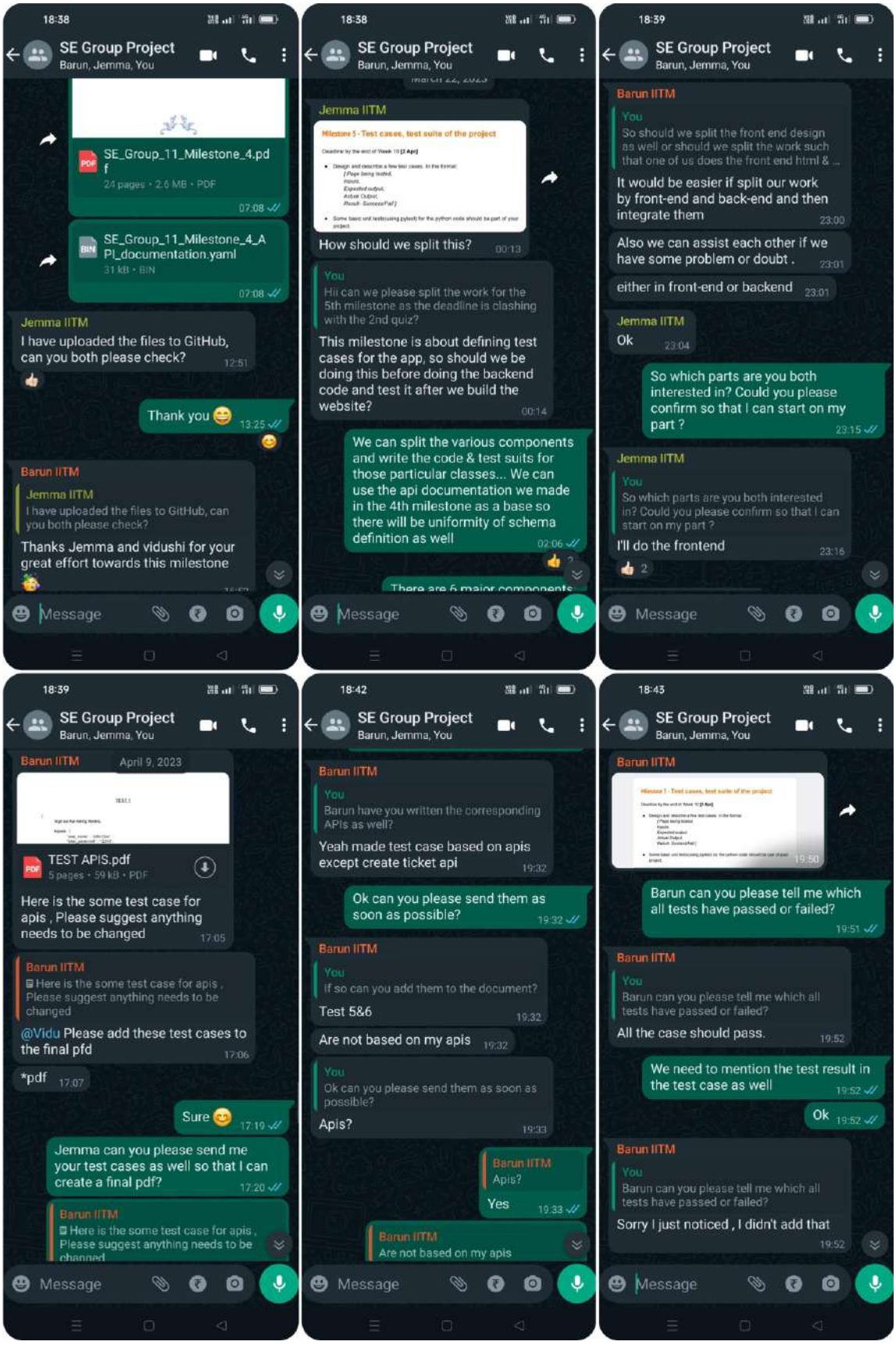
Issue Tracking & Code review

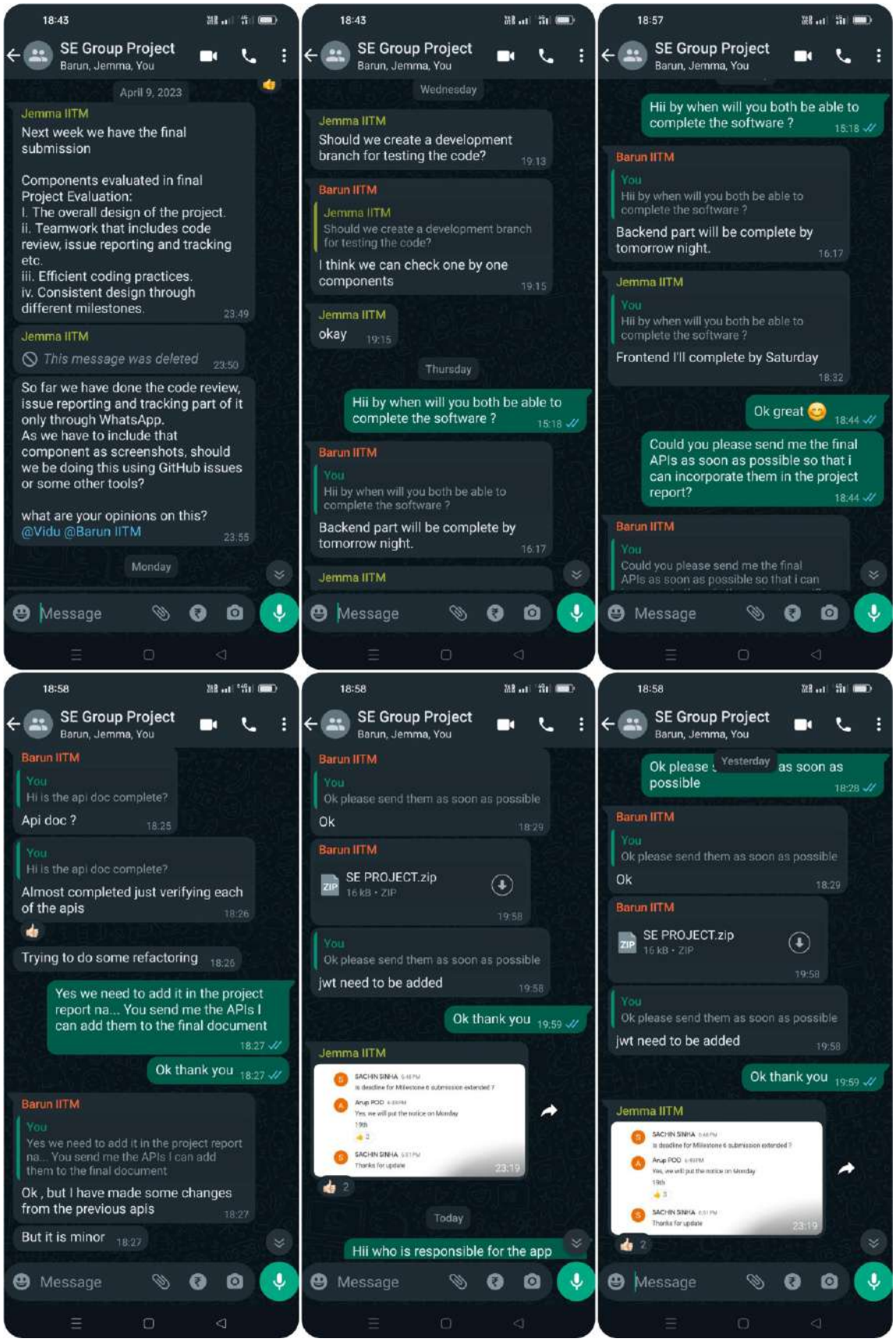
At the beginning of the semester, we created a WhatsApp group which was used as a medium of interaction throughout the project. Some snippets of the conversation have been attached herewith:




18:35
SE Group Project
Barun, Jemma, You
Jemma IITM
You
So does my part in the 2nd milestone conclude or should I contribute to any other part as well?
You may conclude your part with this.Thank you for your time and efforts
23:14
Ok Thank you so much
23:46
February 21, 2023
Barun IITM
Jemma Can you try paper prototype ?
17:18
Actually I'm not okay with drawing .
17:19
Please can you both complete till 2nd Milestone .
17:20
I can do all coding part of the project .
17:20
Jemma IITM
Barun IITM
Message
18:35
SE Group Project
Barun, Jemma, You
February 26, 2023
Jemma IITM
SE_Group_11_Milestone_2_Paper_Prototype.pdf
11 pages • 2.6 MB • PDF
22:56
You
I was thinking we can take rounds while committing/ uploading the pdf on GitHub.... We can start roll number wise.....
@Barun IITM can you upload this time?
22:59
Barun IITM
Jemma IITM
@Barun IITM can you upload this time?
Done
23:46
Thank you @Jemma IITM and @Vidu for your great effort .
23:47
Message
18:36
SE Group Project
Barun, Jemma, You
February 27, 2023
Jemma IITM
https://www.figma.com/file/0C0R7R7ohF8fR2Jadppvl1/Quick-Resolve?node-id=0%3A1&t=mCjf3iLLLqkaQaIY-0
00:03
Jemma IITM
https://www.figma.com/file/0C0R7R7ohF8fR2Jadppvl1/Quick-Resolve?node-id=0%3A1&t=mCjf3iLLLq...
This is the figma file, we can work on this for milestone 3
00:04
2
Hii so sorry i forgot my phone at home last night so couldn't reply... Really great work Jemma
14:02
February 28, 2023
Hii can we split the work for the 3rd milestone? We should start working on it...
13:30
Jemma IITM
Message
18:36
SE Group Project
Barun, Jemma, You
Barun IITM
Dates need to be added
01:15
Barun IITM
01:15
Barun IITM
01:15
Barun IITM
01:15
Please suggest anything need to be changed or to be added
01:16
Hi sorry for the late response
Is this the entire project schedule? I just had 1 suggestions.. shouldn't the spelling of QuickResolve be the same as used in the wireframes?
16:17
Message
18:36
SE Group Project
Barun, Jemma, You
Barun IITM
You
Hi sorry for the late response
Is this the entire project schedule? I just had 1 suggestions.. shouldn't the spellin...
No. Actually in scrum (sprint) only devlopement part is included .
20:40
and for the spelling It is limited to 10 chars only.
20:41
Ok got it! great work Barun
20:49
When do we have to submit?
20:49
Barun IITM
tommrow ?
20:49
I usually forget the dates .
20:50
Barun IITM
tommrow ?
Ya i just checked the portal its on 5th (tomorrow) only
21:06
March 5, 2023
Message
18:36
SE Group Project
Barun, Jemma, You
Barun IITM
Class Diagram
Class diagram.pdf
2 pages • 107 kB • PDF
01:03
Barun IITM
Class diagram.pdf (2 pages)
Please review it .
01:03
Hii could you both please share the documents so that we can submit?
18:27
Jemma IITM
This message was deleted
20:30
You
Hii could you both please share the documents so that we can submit?
Should I convert each screen in this to a pdf or adding the link to figma file in pdf would do?
The protoyping part would be
Message


18:36
SE Group Project
Barun, Jemma, You
March 5, 2023
You
Hii could you both please share the documents so that we can submit?
Should I convert each screen in this to a pdf or adding the link to figma file in pdf would do?
The protoyping part would be possible only with the link
20:33
You
Hii could you both please share the documents so that we can submit?
Design of components:
https://www.figma.com/file/0C0R7R7ohF8fR2Jadppvl1/Quick-Resolve?node-id=63%3A278&t=TKDEC5o9iWYueO1P-1
Prototype view:
https://www.figma.com/proto/0C0R7R7ohF8fR2Jadppvl1/Quick-Resolve?node-id=63%3A278&scaling=contain&page-id=0%3A1&starting-point-node-id=63%3A278
20:54
Jemma IITM
Should I convert each screen in this to a pdf or adding the link to figma file in pdf would do?...
Message
18:37
SE Group Project
Barun, Jemma, You
I think converting it to a pdf would be better... What say?
21:23
Barun IITM
You
I think converting it to a pdf would be better... What say?
Yeah And I think it is also recommended to submit in pdf
21:29
Barun IITM
SE_Group_11_Milestone_3_Class_diagram.pdf
209 kB • PDF
21:30
Jemma IITM
SE Group_11_Design of Components_Milestone 3.pdf
8 pages • 1.1 MB • PDF
Message
18:37
SE Group Project
Barun, Jemma, You
Jemma IITM
SE Group_11_Design of Components_Milestone 3.pdf
8 pages • 1.1 MB • PDF
22:13
Great Work Barun and Jemma
22:16
2
Barun IITM
SE_Group_11_Milestone_3_Sprints_Gantt_chart.pdf
3 pages • 173 kB • PDF
22:16
Should i go ahead with the submission?
22:17
Barun IITM
Message
18:37
SE Group Project
Barun, Jemma, You
March 16, 2023
The api endpoints to be implemented include:
05:08
1. create_ticket
05:08
2. Like_ticket
3. Resolve_ticket
4. Add_ticket_to_faq
5. Report_ticket
6. Delete_ticket
05:10
So we can probably split the endpoints according to the components (Student, support staff & admin) and work on them and then one of us can compile in the end... What say?
05:13
Which parts would you both like to do?
05:14
Jemma IITM
You
So we can probably split the endpoints according to the components (Student, support staff & admin) and work on the...
I'll work on the student part
09:22
Message
18:38
SE Group Project
Barun, Jemma, You
March 16, 2023
Barun IITM
I'll work on support staff api and login & signup .
10:25
Ok great ! Then I'll take up admin api & compiling
14:39
Can you please send me your APIs by 18th so that I can work on it at the earliest? I'll need time to make the final doc
14:41
Jemma IITM
You
Can you please send me your APIs by 18th so that I can work on it at the earliest? I'll need time to make the final d...
Ok
14:46
Barun IITM
You
Can you please send me your APIs by 18th so that I can work on it at the earliest? I'll need time to make the final d...
sure .
18:05
March 18, 2023
Message
18:38
SE Group Project
Barun, Jemma, You
March 18, 2023
Jemma IITM
openapi (1).yaml
5.6 kB • BIN
23:35
Jemma IITM
openapi (1).yaml
Please do suggest changes if needed
23:36
March 19, 2023
Barun IITM
openapi.yaml
3.2 kB • BIN
00:39
Great work Jemma and Barun I had a small suggestion... as per what I remember from the MAD days, I think we should stick to the design components document we made in milestone 3 for uniformity as much as possible.. so I tried to incorporate the documents and created this yaml file.. Could you both please check and provide your feedback?
06:47
Message


18:38
SE Group Project
Barun, Jemma, You
SE_Group_11_Milestone_4.pdf
24 pages • 2.6 MB • PDF
07:08
SE_Group_11_Milestone_4_API_documentation.yaml
31 kB • BIN
07:08
Jemma IITM
I have uploaded the files to GitHub, can you both please check?
12:51
Thank you
13:25
Barun IITM
Jemma IITM
I have uploaded the files to GitHub, can you both please check?
Thanks Jemma and vidushi for your great effort towards this milestone
Message
18:38
SE Group Project
Barun, Jemma, You
Jemma IITM
Milestone 5 - Test cases, test suite of the project
How should we split this?
00:13
You
Hii can we please split the work for the 5th milestone as the deadline is clashing with the 2nd quiz?
This milestone is about defining test cases for the app, so should we be doing this before doing the backend code and test it after we build the website?
00:14
We can split the various components and write the code & test suits for those particular classes... We can use the api documentation we made in the 4th milestone as a base so there will be uniformity of schema definition as well
02:06
Message
18:39
SE Group Project
Barun, Jemma, You
Barun IITM
You
So should we split the front end design as well or should we split the work such that one of us does the front end html & ...
It would be easier if split our work by front-end and back-end and then integrate them
23:00
Also we can assist each other if we have some problem or doubt .
23:01
either in front-end or backend
23:01
Jemma IITM
Ok
23:04
So which parts are you both interested in? Could you please confirm so that I can start on my part ?
23:15
Jemma IITM
You
So which parts are you both interested in? Could you please confirm so that I can start on my part ?
I'll do the frontend
23:16
2
Message
18:39
SE Group Project
Barun, Jemma, You
Barun IITM
April 9, 2023
TEST APIS.pdf
5 pages • 59 kB • PDF
Here is the some test case for apis , Please suggest anything needs to be changed
17:05
Barun IITM
Here is the some test case for apis , Please suggest anything needs to be changed
@Vidu Please add these test cases to the final pfd
17:06
*pdf
17:07
Sure
17:19
Jemma can you please send me your test cases as well so that I can create a final pdf?
17:20
Barun IITM
Here is the some test case for apis , Please suggest anything needs to be changed
Message
18:42
SE Group Project
Barun, Jemma, You
Barun IITM
You
Barun have you written the corresponding APIs as well?
Yeah made test case based on apis except create ticket api
19:32
Ok can you please send them as soon as possible?
19:32
Barun IITM
You
If so can you add them to the document?
Test 5&6
19:32
Are not based on my apis
19:32
You
Ok can you please send them as soon as possible?
Apis?
19:33
Barun IITM
Apis?
Yes
19:33
Barun IITM
Are not based on my apis
Message
18:43
SE Group Project
Barun, Jemma, You
Barun IITM
Milestone 1 - Test cases, test suite of the project
19:50
Barun can you please tell me which all tests have passed or failed?
19:51
Barun IITM
You
Barun can you please tell me which all tests have passed or failed?
All the case should pass.
19:52
We need to mention the test result in the test case as well
19:52
Ok
19:52
Barun IITM
You
Barun can you please tell me which all tests have passed or failed?
Sorry I just noticed , I didn't add that
19:52
Message


SE Group Project
Barun, Jemma, You
April 9, 2023
Jemma IITM
Next week we have the final submission
Components evaluated in final Project Evaluation:
I. The overall design of the project.
ii. Teamwork that includes code review, issue reporting and tracking etc.
iii. Efficient coding practices.
iv. Consistent design through different milestones.
23:49
Jemma IITM
This message was deleted
23:50
So far we have done the code review, issue reporting and tracking part of it only through WhatsApp.
As we have to include that component as screenshots, should we be doing this using GitHub issues or some other tools?
what are your opinions on this?
@Vidu @Barun IITM
23:55
Monday
Wednesday
Jemma IITM
Should we create a development branch for testing the code?
19:13
Barun IITM
Jemma IITM
Should we create a development branch for testing the code?
I think we can check one by one components
19:15
Jemma IITM
okay
19:15
Thursday
Hii by when will you both be able to complete the software ?
15:18
Barun IITM
You
Hii by when will you both be able to complete the software ?
Backend part will be complete by tomorrow night.
16:17
Jemma IITM
You
Hii by when will you both be able to complete the software ?
Frontend I'll complete by Saturday
18:32
Ok great
18:44
Could you please send me the final APIs as soon as possible so that i can incorporate them in the project report?
18:44
Barun IITM
You
Could you please send me the final APIs as soon as possible so that i can
Hi is the api doc complete?
Api doc ?
18:25
Almost completed just verifying each of the apis
18:26
Trying to do some refactoring
18:26
Yes we need to add it in the project report na... You send me the APIs I can add them to the final document
18:27
Ok thank you
18:27
Ok , but I have made some changes from the previous apis
18:27
But it is minor
18:27
Ok please send them as soon as possible
Ok
18:29
SE PROJECT.zip
16 kB • ZIP
19:58
jwt need to be added
19:58
Ok thank you
19:59
Today
Hii who is responsible for the app
Yesterday
Message

Deployment

The final stage of the SDLC life cycle involves delivering the end product to the user.

Deployment is the mechanism through which applications, modules, updates & patches are delivered from developers to users.

We have deployed our application using the Netlify & Replit platform.

Netlify: Quick Resolve

Replit: Quick Resolve

Note: Due to technical difficulties the database couldn't be deployed at the Netlify platform. Hence some functionalities might not work on the given Netlify link.

For best user experience it is advised to zoom in your browser at 110%

Technologies & Tools

The technologies & tools used for the development of the project include:

Tool	Use Case
Jira	Project Schedule
Figma	Framework Design
Swagger API	API Documentation
Storyboardthat	Creating Storyboard
Pytest	Test Driven Developement
HTML / CSS	Front-end Development
Vuejs	Front-end Devlopment
Visual Studio	Coding
Git Hub	Project Submission
Microsoft Word	Project Documentation
Netlify	Project Deployment
Replit	Project Deployment