## Planning

Understand where random questions are selected?

Wanted to implement various conditions in the program.

Firstly, provide id and ratting to each questions so that later we can add conditions on them.

Add start paper button in starting.

Keep total 60 questions in paper, 20 question of each level.

Conditions 1 : start question with begginer level, if user mark correct answer in begginer then next question selected will be of intermediate level, and if marked question is wrong then send him begginer level question again.

Condition 2 : if user is at intermediate level and mark correct answer then send him expert level question. and if marked wrong then send him begginer level question.

Condition 3 : similar to condidtion 1 and 2 user will be at expert level and if answered correvctly then will pass the test and if answered wrong then will go on intermediate level.

soln

search for 3 4 algorithm in typescript to impliment this condition

Algo

Algo 1 – (make various classes for them in microservice and then add every component in its specific class)

questions = [

{id: 1, rating: "beginner"},

{id: 2, rating: "beginner"},

...

{id: N, rating: "expert"}

]

currentLevel = "beginner"

correctAnswers = 0

while currentLevel != "pass" and userContinues():

if currentLevel == "beginner":

question = randomlySelectQuestion("beginner")

else if currentLevel == "intermediate":

question = randomlySelectQuestion("intermediate")

else if currentLevel == "expert":

question = randomlySelectQuestion("expert")

displayQuestion(question)

if userAnswersCorrectly():

correctAnswers += 1

if currentLevel == "beginner" and correctAnswers == 1:

currentLevel = "intermediate"

else if currentLevel == "beginner" and correctAnswers == 2:

currentLevel = "expert"

else if currentLevel == "intermediate":

currentLevel = "expert"

else:

correctAnswers = 0

if currentLevel == "intermediate" or currentLevel == "expert":

currentLevel = "beginner"

if currentLevel == "pass":

displayPassMessage()

else:

displayStopMessage()

put this code in service class :