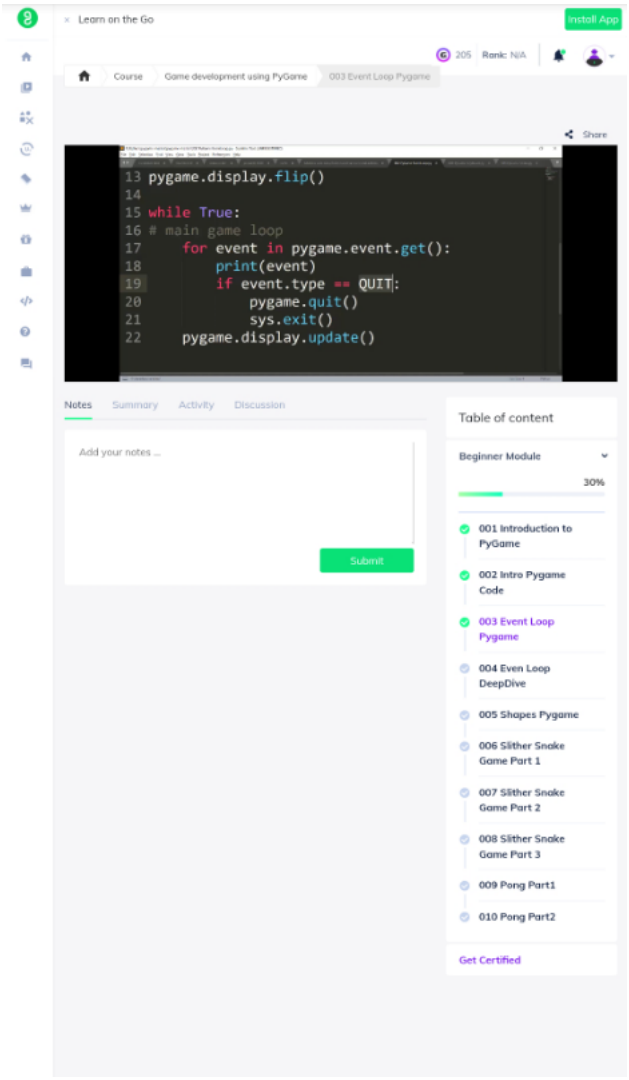
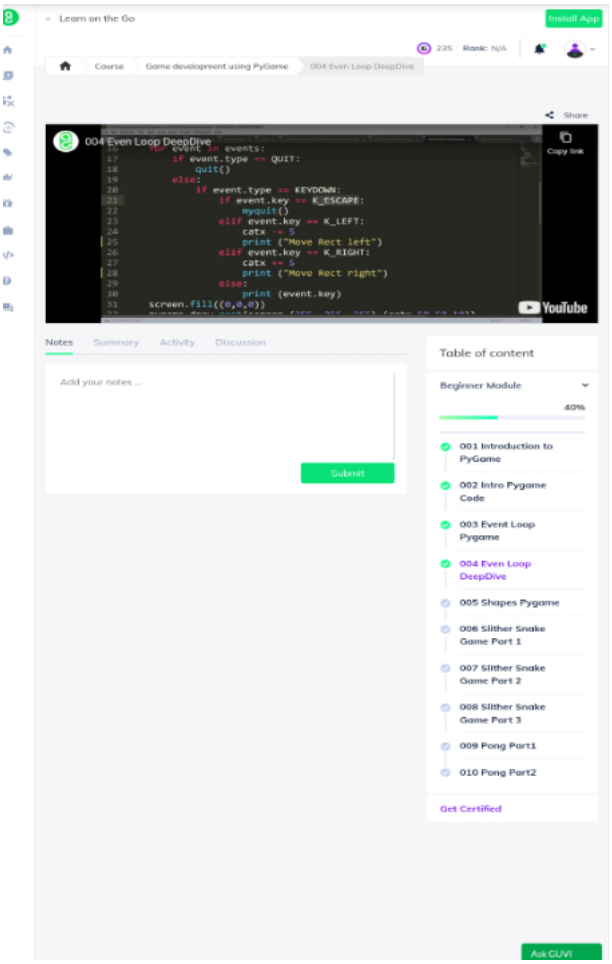
DAILY ONLINE ACTIVITIES SUMMARY

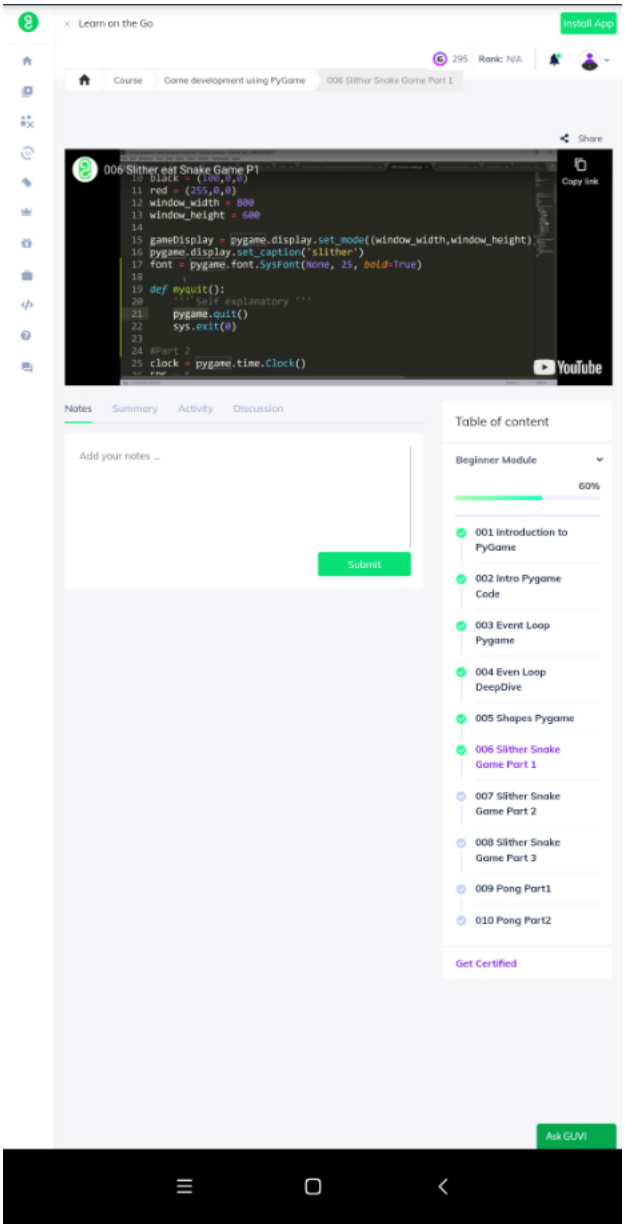
Date:	03-06-2020	Name:	Shaima Abdul Kader
Sem & Sec	VIII Semester & B Section	USN:	4AL16CS087
Online Test Summary			
Subject	-		
Max. Marks	-	Score	-
Certification Course Summary			
Course	Game development using Pygames		
Certificate Provider	Ui Path	Duration	3 Hours
Coding Challenges			
Problem Statement: Find an array of positive integers for the inversion count of array.			
Status: COMPLETED			
Uploaded the report in Github	YES		
If yes Repository name	shaima		
Uploaded the report in slack	YES		

Online Test Details:

Certification Course Details:




Learn on the Go
Install App
235
Rank: N/A
Course
Game development using PyGame
004 Even Loop DeepDive
Share
004 Even Loop DeepDive
Copy link
YouTube
Notes
Summary
Activity
Discussion
Add your notes ...
Submit
Table of content
Beginner Module
40%
001 Introduction to PyGame
002 Intro Pygame Code
003 Event Loop Pygame
004 Even Loop DeepDive
005 Shapes Pygame
006 Slither Snake Game Part 1
007 Slither Snake Game Part 2
008 Slither Snake Game Part 3
009 Pong Part1
010 Pong Part2
Get Certified
Ask GUVI

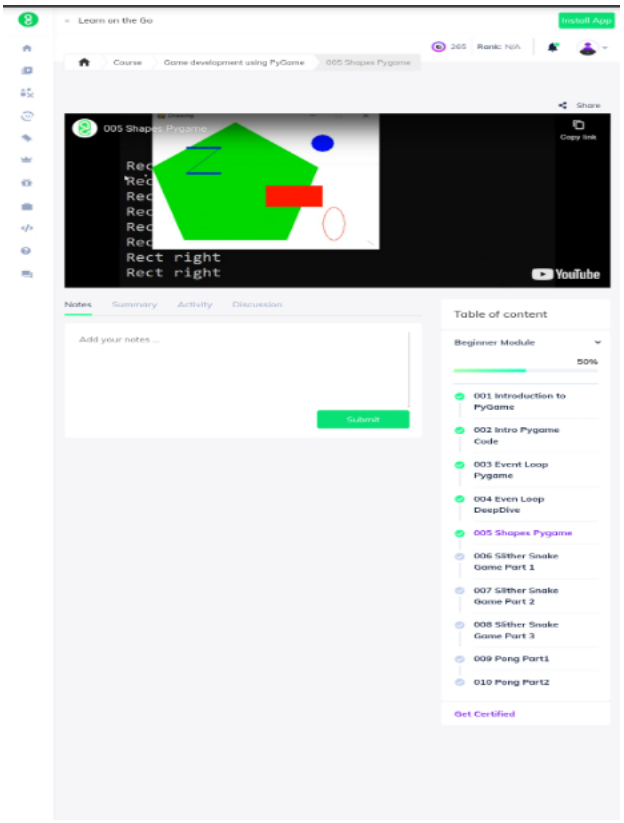


Pygame uses the Simple DirectMedia Layer (SDL) library,[a] with the intention of allowing real-time computer game development without the low-level mechanics of the C programming language and its derivatives. This is based on the assumption that the most expensive functions inside games can be abstracted from the game logic, making it possible to use a high-level programming language, such as Python, to structure the game.

Other features that SDL doesn't have include vector math, collision detection, 2d sprite scene graph management, MIDI support, camera, pixel-array manipulation,

transformations, filtering, advanced freetype font support, and drawing.

Applications using pygame can run on Android phones and tablets with the use of pygame Subset for Android. Sound, vibration, keyboard, and accelerometer are supported on Android.



Coding challenges online details :

```
#include<stdio.h>


int getInvCount(int arr[], int n)
{
   int inv_count = 0;
   for (int i = 0; i < n - 1; i++)
      for (int j = i + 1; j < n; j++)
```

```
            if (arr[i] > arr[j])
                inv_count++;


    return inv_count;
}


int main(int argv, char** args)
{
    int arr[] = { 2,4,1,3,5 };
    int n = sizeof(arr) / sizeof(arr[0]);
    printf(" Number of inversions are %d \n", getInvCount(arr, n));
    return 0;
}
```