code.sprint MT

PREPARATION BOOKLET

Developers 2022

DIRECTORATE FOR LEARNING &
ASSESSMENT PROGRAMMES

Table of Contents

Intro

Welcome to the code.sprint 2022 competition. During this two-day competition, you will be flexing your coding and UI/UX design muscles to create an app!

This booklet provides some resources to help you prepare. In addition, make sure to read carefully the Rules & Regulations and the Terms & Conditions on the code.sprint website

We wish you the best of luck in the competition!

From the code.sprint developers category judging panel 😊

Technical Overview

During the competition, you will be designing and building an app.

The judging panel has created a reference implementation of this app in 16 hours. This is to ensure that the task given is possible within the timeframe allocated.

Platform

Your app can run on any platform. This includes Windows, macOS, Linux, Android, iOS or the web. We recommend a web-based platform be developed; however, this will not affect points awarded and is simply for the app to be usable on as many platforms as possible.

Development Environment

You are free to use any development environment you wish. However, do note that you must provide all source code to the judging panel, as well as instructions on how to setup an environment which allows the judges to run the solution on their workstation.

Preparation Resources

The following resources will help you to prepare for the task. Remember you can use any technology you are familiar with.

Topics you must know (This is the important stuff)

1. Building an application with a graphical user interface, either on desktop, mobile or web (you will be able to choose any platform).
2. Interacting with REST APIs.
3. Capturing images and videos from a webcam/camera connected to the system.
4. Loose coupling, function cohesion and a modern programming paradigm of your choice.

Recommended Reading

1. Computer Vision
    a. https://en.wikipedia.org/wiki/Computer_vision
    b. https://www.ibm.com/topics/computer-vision
2. Webcam Capture (Links provided for common languages. Check links for your preferred language if not below)
    a. Python: https://www.geeksforgeeks.org/python-opencv-capture-video-from-camera/
    b. Java: http://webcam-capture.sarxos.pl / https://www.baeldung.com/java-capture-image-from-webcam
    c. Ruby: https://github.com/TyounanMOTI/rb_webcam
    d. Go: https://medium.com/learning-the-go-programming-language/realtime-video-capture-with-go-65a8ac3a57da
    e. JavaScript: https://makitweb.com/how-to-capture-picture-from-webcam-with-webcam-js/

Other

1. You will need a computer with a built-in or connected webcam.
2. Should you decide to build a cloud-based solution, it is advisable to have created your cloud account in advance. Any cloud provider can be used (AWS, Azure, GCP etc).

Judgement Criteria

Your submission will be given a maximum of 155 points. The criteria by which points are awarded are detailed below. **Note that you do not need to achieve all the criteria**, however, the more criteria you achieve, the greater your chances of winning!

Criterion	Notes	Maximum Points
Core Functionality		
Design Brief 1**		**20**
Design Brief 2**		**20**
Design Brief 3**		**20**
Design Brief 4**		**20**
Design Brief 5**		**10**
Design Brief 6**		**10**
UI/UX		
Neat/Aesthetically pleasant user interface	Rather than 'flair', we are looking for a neat, organized and functional UI	**10**
App is easy to use	The user should not need a manual to use the app	**5**
Experience**		**15**
Code Quality		
Code is organized into packages/modules/units etc.		**5**
Separation between presentation and logic layers	For example, using a REST API model	**10**
Consistent and correct use of a programming paradigm	Such as OOP, AOP, functional etc.	**5**
Function cohesion	Functions should be kept small, and do one thing, without being too dependent on other functions	**5**
Inline documentation	i.e. comments	**5**
Maintainable code	Ex: use of abstract classes, interfaces, function prototypes etc. Depending on the programming paradigm chosen	**5**
Additional Functionality/Features		
Sample data is persistently stored**		**5**
Additional features	Additional features over and above the design brief will be graded, up to a maximum of 15 points	**15**

*** These will be revealed in the task booklet.*

Submission Criteria

At the end of the time allocated to this competition, you must submit your code to the judging panel as instructed during the competition. The code, including all assets and other resources, must be submitted as a folder or compressed archive.

You will also be required to demonstrate your application running.