

code.sprint^{MT}

PREPARATION BOOKLET

Open Category

2023



GOVERNMENT OF MALTA
MINISTRY FOR EDUCATION, SPORT,
YOUTH, RESEARCH AND INNOVATION
DIRECTORATE FOR LEARNING AND ASSESSMENT PROGRAMMES



Table of Contents

Hello!	2
Technical Overview	3
Platform.....	3
Development Environment	3
Preparation Resources	3
Topics you must know (<i>This is the important stuff</i>)	3
Recommended Reading.....	3
Other	4
Judgement Criteria	5
Submission Criteria	5

Hello!

Dear CODESPRINT 2023 Competitors,

Welcome to the exhilarating Code.Sprint 2023 competition! Get ready to unleash your coding prowess, showcase your infrastructure expertise, and demonstrate your exceptional UI/UX design skills as you dive into the world of app development.

This comprehensive preparatory booklet has been carefully crafted to provide you with all the essential information you need to excel in the competition. It offers an resources and guidance to help you prepare effectively.

The judging panel is comprised of experienced professionals who will assess your submissions based on various criteria, including functionality, innovation, user experience, and technical implementation. We have full confidence in your abilities and anticipate witnessing your exceptional creations.

We encourage you to embrace the spirit of healthy competition, channel your creativity, and leverage your skills to develop an app that truly stands out. Make the most of this incredible opportunity to learn, grow, and challenge yourself in the world of coding.

As you embark on this thrilling CODESPRINT journey, we extend our best wishes to each one of you. May your dedication, hard work, and talent shine through, propelling you towards success in the competition.

From the Code.Sprint 2023 Open Category judging panel.

Please note that for comprehensive information regarding the contest schedule, rules & regulations, frequently asked questions, and terms & conditions, we kindly request you to visit our official website at www.codesprintmalta.edu.mt. You will find all the necessary details to ensure your participation in the competition is well-informed and in compliance with the guidelines.

Technical Overview

Throughout the competition, your task will involve designing, and constructing an app.

To demonstrate the feasibility of the given task within the designated timeframe, the judging panel has prepared a reference implementation of this app in just one working day. This serves as a benchmark to ensure that the project can be successfully accomplished within the allocated time.

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. Developing an application featuring a graphical user interface (GUI) on a platform of your choice, including desktop, mobile, or web environments.
2. Creating a simple REST API.
3. Launching a virtual machine on a remote server or in a cloud environment.
4. Creating a pre-defined template for launching a virtual machine.
5. Creating a pre-defined connection file for RDP and VNC connections.
6. Loose coupling, function cohesion and a modern programming paradigm of your choice.

Recommended Reading

- RDP File Format: <https://www.donkz.nl/overview-rdp-file-settings/>

- VNC File Format:
https://www.reddit.com/r/sysadmin/comments/1qh9fk/batch_creation_of_vnc_files
- Launch Templates:
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/create-launch-template.html>
<https://docs.aws.amazon.com/autoscaling/ec2/userguide/launch-templates.html>
- EC2 Instance:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html
- API Gateway:
<https://docs.aws.amazon.com/apigateway/latest/developerguide/getting-started.html>
- Amazon Machine Images (AMIs)
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html>

Azure

- ARM Templates:
<https://learn.microsoft.com/en-us/azure/azure-resource-manager/templates/overview>
- Azure Virtual Machines:
<https://learn.microsoft.com/en-gb/training/modules/create-windows-virtual-machine-in-azure/>
- Azure API Management:
<https://learn.microsoft.com/en-GB/shows/Hello-World/Hello-World-API-Management#time=04m05s?culture=en-gb&country=gb>
- VM Images:
<https://learn.microsoft.com/en-us/azure/virtual-machines/capture-image-portal>

Other

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 185 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		
[1] Design brief 1**		10
[2] Design brief 2**		30
[3] Design brief 3**		10
[4] Design brief 4**		20
[5] Design brief 5**		20
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	The User Experience is up to you; however, more specific details will be given in the task booklet during the competition.	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	For example, the use of abstract classes, interfaces, function prototypes etc. Depending on the programming paradigm chosen	5
Additional Functionality/Features		
Sample data is persistently stored	** available should be stored in a file or database rather than being hardcoded.	5
**	**	10
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. 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 during a brief VIVA session.