codesprint

NATIONAL CODING COMPETITION

Preparation Booklet

Open Category 2026

POWERED BY me direct

GOVERNMENT OF MALTA
MINISTRY FOR EDUCATION AND SPORTS
DIRECTORATE FOR STEM AND VET PROGRAMMES

× icecampus

Table of Contents

Hello!

Welcome to the CodeSprintMT 2026 competition. During this competition, you will be flexing your coding, infrastructure, and UI/UX design muscles to create an app!

This booklet provides a brief overview of the rules and regulations of the competition, as well as providing some resources to help you prepare.

We wish you the best of luck in the competition!

From the CodeSprintMT Open 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 one working day. This is to ensure that the task given is possible within the timeframe allocated. You will be given access to a demonstration of this implementation on competition day.

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** programming language you wish to create your solution. However, do remember that the solution must run on the judge's computers, and that you must provide both a binary/executable solution, as well as source code.

Preparation Resources

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

The Mastercard Sandbox

- Creating a Mastercard Sandbox Account
  https://developer.mastercard.com/donations/documentation/quick-start-guide/#generate-credentials-and-create-a-sandbox-project
- Mastercard Donate
  https://developer.mastercard.com/product/mastercard-donate/
- Donate API - documentation
  https://developer.mastercard.com/donations/documentation
- Donate API - API reference
  https://developer.mastercard.com/donations/documentation/api-reference/
- Donate API - API basics
  https://developer.mastercard.com/donations/documentation/api-basics/
- Mastercard Postman workspace (call the APIs before writing code)
  https://www.postman.com/mastercard/workspace/mastercard-developers/overview

Payments & Contactless Fundamentals

- EMVCo
  https://www.emvco.com/
- Stripe Docs: Tap to Pay
  https://docs.stripe.com/terminal/payments/setup-reader/tap-to-pay
- Google Pay: Tap to Pay overview
  https://developers.google.com/pay/issuers/tap-to-pay

Technical Resources

- Flutter docs
  https://docs.flutter.dev/
- React Native docs
  https://reactnative.dev/docs/getting-started
- Android NFC basics
  https://developer.android.com/develop/connectivity/nfc
- Android Host Card Emulation (HCE)
  https://developer.android.com/develop/connectivity/nfc/hce

Web Development Resources

- React
  https://react.dev/learn

- Next.js docs
  https://nextjs.org/docs
- Vue
  https://vuejs.org/guide/introduction.html
- Tailwind CSS
  https://tailwindcss.com/docs
- Recharts
  https://recharts.org/
- Chart.js
  https://www.chartjs.org/docs/latest/
- TanStack Table
  https://tanstack.com/table/latest

Backend Development Resources

- Node.js docs
  https://nodejs.org/en/docs
- Express
  https://expressjs.com/
- FastAPI (Python)
  https://fastapi.tiangolo.com/
- ASP.NET Core
  https://learn.microsoft.com/en-us/aspnet/core/
- REST API design guidelines (Microsoft)
  https://github.com/microsoft/api-guidelines

Auth & Role-Based Access

- Auth0 docs
  https://auth0.com/docs
- Clerk docs
  https://clerk.com/docs
- jwt.io — Introduction to JWTs
  https://jwt.io/introduction
- Auth0: Role-Based Access Control (RBAC)
  https://auth0.com/docs/manage-users/access-control/rbac

Database

- PostgreSQL docs
  https://www.postgresql.org/docs/
- Supabase
  https://supabase.com/docs

- Prisma ORM
  https://www.prisma.io/docs

Webhooks & Real-Time Dashboards

- webhooks.fyi (vendor-neutral best practices: signing, retries, idempotency)
  https://webhooks.fyi/
- ngrok docs (expose localhost to receive sandbox webhooks during dev)
  https://ngrok.com/docs
- Socket.IO (push live updates to the dashboard)
  https://socket.io/docs/v4/

Accessibility, Internationalisation & Multi-Currency

- WCAG at a glance (W3C/WAI) — large-text & high-contrast requirements
  https://www.w3.org/WAI/standards-guidelines/wcag/
- WebAIM Contrast Checker (verify high-contrast mode in seconds)
  https://webaim.org/resources/contrastchecker/
- i18next
  https://www.i18next.com/
- FormatJS / react-intl
  https://formatjs.io/docs/getting-started/installation/
- MDN: Intl.NumberFormat (correct multi-currency formatting)
  https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Intl/NumberFormat
- ISO 4217 currency codes
  https://www.iso.org/iso-4217-currency-codes.html

Diagrams, Readme, Exports

- C4 model (architecture diagrams)
  https://c4model.com/
- Mermaid (diagrams as code)
  https://mermaid.js.org/
- Excalidraw
  https://excalidraw.com/
- draw.io
  https://app.diagrams.net/
- makeareadme.com
  https://www.makeareadme.com/
- Papa Parse (CSV export)
  https://www.papaparse.com/
- pdfmake (PDF export)
  http://pdfmake.org/

- jsPDF (PDF export)
  https://github.com/parallax/jsPDF

AI-Assisted Coding

Intro

The aim of the CodeSprintMT competition is to reward developers who can solve problems quickly and creating prototypes and proof-of-concept apps for real-world scenarios. In 2026, that will inevitably involve the use of AI-assisted development.

Ok, so can I use AI?

Yes. In fact, it is expected that you will.

Do I have to comment sections created by AI for attribution?

No. AI-created code is a result of your prompt and existing work. The AI does not "own" any code it generates, you do. Commenting sections created or enhanced with AI is impractical and results in messy code files.

Will the use of AI be penalised?

No, if you understand what the AI has created for you. During the VIVA of your submission, you may be asked to explain certain parts of your code. If you can fully explain what your code is doing, there is no penalty for using AI, or indeed the need to disclose which parts of your code were written by AI.

Note, however, that you should always verify any code generated by AI (or indeed, code samples you retrieve via traditional means) for correctness and security. If your code is messy or insecure, you will be penalised, regardless of who or what wrote it!

Judgement Criteria

Your submission will be given a maximum of 345 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		
[M1.1] …	*Something related to branding***	**10**
[M1.2] …	*Something related to available options available to user***	**10**
[M1.3] …	*Something related to user flow through app***	**20**
[M1.4] …	*Something related to user confirmation***	**20**
[M1.5] Multi-Currency	*Ability to support at least three currencies, with EUR as the primary*	**15**
[M1.6] Graceful Error Handling	*Validation and error handling for incorrect input or exceptions*	**15**
[M1.7] Accessibility	*Availability of text size controls and colour scheme swap for high-contrast*	**10**
[M2.1] Role-Based Access Control	*Logging into the admin tool should allow for increasing functionality based on role*	**20**
[M2.2] Create & Manage …	*Ability to create, edit and delete something** with fields required*	**20**
[M2.3] Live Dashboard	*A dashboard showing live data from activities** carried out by the users*	**20**
[M2.4] Transaction Ledger	*List of all transactions carried out in app. Filterable and sortable*	**15**
[M2.5] CSV & PDF Export	*Transaction ledger exportable as CSV or PDF*	**10**
[M2.6] Reconciliation View	*Ability to match transaction records from an API** to those recorded from the app*	**10**
[M2.7] Audit Log	*Log of every admin action*	**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**

Responsiveness	The app should be usable on both desktop and mobile screens	**10**
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		
[S1.1] …	*Something related to extra actions following a user action***	**10**
[S1.2] …	*A UI prototype for an additional feature***	**10**
[S1.3] …	*A more functional prototype for a UI feature***	**20**
[C1.1] Mobile App	Prototype of mobile app	**15**
[C2.1] Multilingual UI	Support for translations of the UI, with at least one partial translation to another language	**15**
[C3.1] Real-time dashboard	Admin dashboard updated in real time	**10**
Additional features	*Used as a tie-break. See below.*	

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

Additional Features

Should you wish to add features to your solution which are over-and-above the requirements and suggestions given in the task, you are free to do so. However, please note that **these will not be graded**.

Additional features are only considered in the unlikely event that two submissions receive the exact same grade. In this case, the winner between the two will be decided by a VIVA as well as additional features provided, and is at the sole discretion of the judging panel.

Submission Criteria

Please read this section **VERY CAREFULLY**.

You must create a demonstration video as part of your submission. This video must showcase EACH and EVERY feature you have implemented in your solution. In your video, make sure you go over all of the functionality of both the app and any backend created. Ideally, the video should be narrated by yourself.

FAILURE TO SUBMIT A VIDEO WILL RESULT IN IMMEDIATE DISQUALIFICATION

ANY FEATURES IMPLEMENTED IN YOUR SOLUTION WHICH ARE NOT DEMONSTRATED IN THE VIDEO WILL NOT BE GRADED. THIS INCLUDES OPTIONAL (NON-CORE) OR ADDITIONAL FEATURES.

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 through the link provided by the organisers at the end of the competition.

Remember to include your demonstration video in the submission.

Rules and Regulations

1. CodeSprintMT is open to anyone residing within the Maltese Islands who codes - whether as a student, graduate, working professional, educator, hobbyist or self taught coder. Participation is confirmed upon successful registration and formal acceptance by the organisers, at their sole discretion.
2. Participants are free to use any tech stack of their choice, including programming languages, frameworks, libraries, and tools they are most comfortable with.
3. The session will be held in-person at ICE Campus (Zebbug) using a bring-your-own-device model. Internet access and an optional extended monitor will be provided. Participants are responsible for ensuring their device is fully functional, pre-loaded with any required software or tools, and equipped with the necessary cable to connect to the extended monitor.
4. All participants will work individually (this is a solo challenge). Strict security protocols will be in place throughout the competition. Any attempt to plagiarise or receive help from another person will result in immediate disqualification.
5. Participants are expected to maintain a respectful environment during the session. This includes keeping noise to a minimum, avoiding unnecessary conversation, and not distracting others. Silence is to be observed throughout the competition room unless speaking with an organiser or judge.
6. Participants are allowed to use AI tools responsibly as part of their workflow. You are not required to comment or attribute AI-generated sections, but you must understand your code and may be asked to explain it during the viva. Messy, insecure, or unverifiable code will be penalised. For full AI usage guidelines, refer to the AI-Assisted Coding section in this document.
7. During the session, it is mandatory that participants:
    a. maintain healthy habits and take short breaks,
    b. save and back up their work regularly throughout the session,
    c. follow all instructions given by organisers and invigilators,
    d. submit their task, including the completed solution and demonstration video, by 5.00PM.
8. Following the competition, all submissions will be reviewed by the judging panel. The top 10 ranked submissions will be shortlisted for a VIVA, held online a couple of days after the competition. Shortlisted participants will be notified by email and given a scheduled time slot.
9. Failure to submit a demonstration video will result in immediate disqualification. Failure to demonstrate a feature in the demonstration video will mean this feature will be ignored by judges during grading.
10. Additional features implemented in a participant's solution which are over-and-above the requirements mentioned in the task booklet will not be graded, unless there is a perfect tie between two participants. In this case, the tie will be resolved via a

combination of participant VIVA and any additional features implemented. This is at the sole discretion of the judging panel.

11. The competition area will be monitored by CCTV throughout the session for security and integrity purposes. Footage is retained only for as long as necessary for these purposes.
12. Photographs and video recordings will be taken during the event for marketing and recap purposes. Participants who do not wish to appear in marketing content may notify the organisers before the competition starts and reasonable steps will be taken to accommodate this.

Terms and Conditions

1. **Right to cancel or modify**
   CodeSprintMT reserves the right to cancel, suspend or modify the challenge if any problem prevents the challenge from running as planned.
2. **Release**
   All participants agree to release and hold CodeSprintMT harmless from and against any claim associated with the challenge/event.
3. **Limitation of liability**
   CodeSprintMT is not responsible and cannot be held liable for technical errors or other things that may prevent the challenge from running as planned.
4. **Right to substitute**
   CodeSprintMT reserves the right to substitute the awarding prizes with another prize if the advertised prize is not available.
5. **Permission to record online VIVA sessions**
   CodeSprintMT reserves the right to record online session/s which will be used as evidence for grading purposes only.
6. **Permission to store participants' work**
   CodeSprintMT reserves the right to store the work submitted by the participants to up to 10 years for records purposes.
7. **Disqualification**
   You may be disqualified from CodeSprintMT contest and forfeit any prizes you may be eligible to receive if the organizer reasonably believes that you have attempted to undermine the legitimate operation of the contest according to the Contest Rules & Regulations. You may report violation of these terms by another contestant by contacting us.