LifeLines
Software that saves lives

HACKATHON '24

CHANGEMAKERS' GUIDE

Table of Contents

A. Lifelines Introductions:

About the Platform

The first of its kind in Qatar, LifeLines is a student-led humanitarian innovation platform at Carnegie Mellon University Qatar that brings together students, educators, and industry professionals to build solutions for mitigating the effects of humanitarian crises on societies around the world, empowering the next generation of responsible professionals and entrepreneurs through hands-on collaborative learning.

What we do

- ***Educate:***
    - We support students on different paths with informed awareness about collective responsibility, using hands-on practice guided by experts at one of the top universities in the world
- ***Create:***
    - We engage rising changemakers and harness technology to tackle the challenges of humanitarian crises on the ground and within the digital sphere
- ***Empower:***
    - We enable humanitarian progress by highlighting promising work that inspires and engages contributors at the frontline of global responsibility

Programs

- Annual Hackathon
- Interdisciplinary Student Workshops
- Open-source, experimental, and research projects
- Online Content and Publications

B. LifeLines Hackathon '24

In an era where the world is increasingly challenged by crises like natural disasters and conflicts, the role of technology in humanitarian aid has become more crucial than ever. This year alone, we've witnessed the catastrophic earthquake in Turkey, the devastating floods in Libya, another earthquake in Morocco, and the war in Gaza, not to forget the decade of war in Syria. These events have consistently placed humanitarian aid at the forefront of global attention.

At LifeLines, we are not just witnesses to these events; we are a community of potential changemakers. Educated in one of the world's most diverse and dynamic academic environments, we possess a unique blend of skills and knowledge. It's time to channel these resources into something impactful.

We recognize that students like ourselves are the biggest force of change, and we invite you to join us in this endeavor. The LifeLines Hackathon '24, taking after our slogan "Software that Saves Lives," is not just a competition; it's a call to action. We are not merely students and faculty; we are innovators and humanitarians, ready to make a difference.

1. Hackathon Themes

This year, teams are invited to create software that saves lives by serving one of the three following domains:

Crisis Management

In the face of unexpected challenges and emergencies, participants in the Crisis Management theme aim to devise solutions that boost our response, recovery, and adaptability during crises. Whether addressing natural disasters, public health emergencies, or unforeseen events, your goal is to leverage technology and creativity to develop software that empowers communities and organizations in managing crisis situations.

Here are some examples of problems you could work on:

Please note that these are just examples. We are expecting you to go far and beyond in your creative process.

- Develop technologies and tools that aid displaced civilians and individuals in danger.
- Create solutions to support and enhance the efforts of rescue workers on the ground.
- Design applications focused on managing and maintaining critical infrastructure, including water supply, electricity, and transportation systems.
- Develop tools to optimize the allocation and distribution of resources in humanitarian aid situations.

Health and Education

In the Health & Education theme, participants are invited to dive into the intersection of two vital sectors—healthcare and education. Your mission is to envision solutions that bridge gaps, improve access, and enhance outcomes in these essential areas. From innovative health technologies to educational platforms that empower learners, your challenge in this theme is to contribute to a healthier, more knowledgeable society by leveraging the power of technology.

Here are some examples of problems you could work on:

Please note that these are just examples. We are expecting you to go far and beyond in your creative process.

- Develop tools that make education more accessible in crisis zones and ensure the continuity of education.
- Develop tools and platforms that support educators in crisis zones, including training modules and resources to help them adapt to challenging teaching environments.
- Create innovations that assist health workers in providing effective medical care in humanitarian situations.
- Develop tools and applications that empower civilians to administer essential medical care to themselves in situations where accessing professional medical aid is challenging or not immediately possible.

Media and Awareness

In the Media & Awareness theme, your objective is to use technology to amplify voices, share information, and drive positive change. Your challenge in this theme is to explore inventive ways to raise awareness on critical issues, promote inclusivity, and empower communities. Your task involves crafting solutions that redefine how we engage with and consume media for a more informed and connected world.

Here are some examples of problems you could work on:

Please note that these are just examples. We are expecting you to go far and beyond in your creative process.

- Develop technologies to monitor and ensure the accuracy of information on social media during crises.
- Develop solutions to support safe and effective reporting by journalists in humanitarian situations.
- Design innovative tools for educational outreach about the cause, providing insights into its history, impact, and significance.
- Develop impactful tools to raise awareness about the cause, encouraging engagement in charitable activities, activism, and more.
- Develop tools to aid information sharing within crisis zones when networks are disconnected.

2. Team Roles

In this hackathon, an ideal team formation will consist of 2 Hackers and 2 Specialists. This is done to create the best balance between technical capability and expert knowledge of the domain at hand. The two roles are described as follows:

Hackers

Hackers in this hackathon are creative problem-solvers who apply their coding skills and technical knowledge to develop innovative solutions in the realm of humanitarian aid. They are the driving force behind the software development process, turning abstract ideas into tangible applications. Their role is crucial in conceptualizing and executing projects that align with their chosen Theme. Hackers are expected to be versatile, able to work collaboratively in teams, and passionate about using technology as a tool for positive change in times of crisis.
- Each team should have 2 hackers at minimum, 3 hackers at maximum
- You can find more details on the role of the hacker in the development guide section

Specialists

Specialists in this hackathon are subject matter experts who bring in-depth knowledge and insights specific to the domains of Crisis Management, Health and Education, and Media and Awareness. They provide the necessary context and expertise that guide the technological solutions being developed. Specialists' role is to ensure that the solutions proposed are not only technically sound but also practically relevant and impactful in real-world scenarios. They play a pivotal role in bridging the gap between technology and its application in humanitarian scenarios.
- Each team should have 1 specialist at minimum, 2 at maximum
- You can find more details on the role of the specialist in the development guide section

*Please note that roles are not concrete and very well **may overlap**. For example, a Hacker may contribute greatly to the solution's pitch, and a specialist may work on a large aspect of the code. Roles are simply titles that suggest a structure for task division and help you create more balanced teams.*

3. Schedule

The hackathon is scheduled as follows:

Day & Date	Time	Event	Location
Day 0: Wed, 7th Feb	17:30	Refreshments & Registration	Assembly Area
	18:00	Welcome	Assembly Area
		Remarks by Dean Khaled	Assembly Area
	18:30	Theme-Specific Introductions	Assigned Rooms
	19:00	Team Formation Session	Assigned Rooms
	20:00	End of Welcome Session	
Day 1: Fri, 9th Feb	09:00	Registration + Giveaway Distribution	Walkway
	09:00	Breakfast	Foodcourt
	10:00	Kickoff!	
	14:00	Lunch	Foodcourt
	18:00	Dinner	Foodcourt
Day 2: Sat, 10th Feb	09:00	Breakfast	Foodcourt
	10:00	Submission Closes	
	10:30	Judging Round One Commences	Assigned Rooms
	12:00	Break + Setup for Showcase	Walkway.
	12:15	Projects Showcase	Walkway
	13:20	Closing Ceremony Begins	Assembly Area
		Keynote Speeches by Dean Trick & Dean Khaled	
	13:50	Judging Round Two Commences	
	14:40	Awards Ceremony	Assembly Area
	15:00	End of Awards Ceremony	

4. Rules & Guidelines

Coding Integrity:
- All projects must be coded only during the given 24 hours. Any frameworks, libraries, or external code extracted from sources can be used as long as it is given proper credit. Any team that breaks this rule will be automatically suspended from the competition.

Attendance Confirmation:
- Each participant must confirm their attendance by being present on campus to sign in on Wednesday, Feb 7th, 5:45 pm.

Dedicated Workspace and mentorship confirmation:
- Each team will be allocated a room that will act as their workspace during the 24 hours. Each team is expected to remain on campus in their room until the ideation phase has ended and approval of the idea has been received by at least one mentor. .

Campus presence:
- Teams are free to leave campus after the ideation phase but must return by 9am on Saturday 10th in time for the judging rounds to begin.

Cooperation with staff:
- Teams must cooperate with the hackathon's volunteers, mentors, and executives during the event, and make sure to follow their instructions carefully.

Collaboration Guidelines:
- Teams are allowed to discuss their projects, but are not allowed to work together, even if they fall within the same theme/challenge. Remember that each team will be evaluated separately.

5. Mentor Support

Our dedicated mentors bring a wealth of knowledge to guide and support you in various fields.

Mentor Fields:
- Programming
- Business Innovation
- Health
- Education
- Media & Awareness
- Crisis Management

Mentor Roles:
- Guide hackathon participants to ensure adherence to project plans.
- Utilize their expertise to assess progress and offer guidance during challenges.
- Dedicate their time to significantly contribute to the success of participants' projects.

Mentor Availability:
- Friday, Feb 9 from 1:30 till 4:30 PM
- Friday, Feb 9 from 5:00 till 8:00 PM
- Overnight (Friday to Saturday) - (Note: Overnight availability might not always be guaranteed)
- Saturday, Feb 10 from 8:30 till 10:30 AM

How to Get Help from Mentors:
- Mentors will visit all teams during different times in the hackathon to assess progress and offer guidance.
- Use this opportunity to discuss, ask questions, and receive feedback.
- After mentors' rotations, teams can reach out to mentors for further assistance on **Discord**.
- If you need mentorship overnight, ask on discord, there might be mentors to help either on zoom or in-person!

6. Progress Checkpoints

Throughout the hackathon, there will be multiple progress checkpoints to make sure that teams are on track. Checkpoints are **not deliverables**, and you are **not required to follow this outline**. They are simply there to guide you through the process. Again, a *suggested* structure.

By each checkpoint, teams should ideally have done the following:

Checkpoint	Time	Objectives
Checkpoint 1: Ideation	Fri, 12pm	- Identified the market gap to tackle. - Decided on a solution (project idea). - Ran the aforementioned details by multiple. mentors for feedback and approval.
Checkpoint 2: Planning	Fri, 3 pm	- Decided the scope of the final deliverable. - Designed the architecture of/plan for the code. - Created an execution plan with a timeline and task delegations.
Checkpoint 3: Preliminary Code	Fri, 9 pm	- Made decent progress in terms of code: 200-400 lines is typically a good indicator. - Not necessarily a working demo, but useful code that will be built on in the final version. - Created early versions of the business model, slides, and pitch.
Checkpoint 4: Final Touches	Sat, 9:30am	- Finalized a working proof of concept: - Proof of concept needs not be a full-fledged project; it needs to clearly show feasibility of the team's solution. - Could be a small scale version, a preliminary model, the core functionality of the software without interface, etc. - Completed all necessary preparations of the presentation (ie, slides, business model, pitch, and demo).

7. Submission & Evaluation

To submit your project, **one member from each team** needs to complete the submission process. Please use the appropriate form link corresponding to your project's theme:

Crisis Management	Health & Education	Media & Awareness
https://forms.gle/j8zuh7SQPf2eqATA7	https://forms.gle/8y6P8bGhFkF6u6tVA	https://forms.gle/ZdaVWJvNwA4SAQcm9

For your submission, please ensure to include the following components:

1. **Public GitHub Repository Link [URL]:**
   a. Your submission should include a comprehensive GitHub repository containing all elements of your project. This encompasses the main project files as well as any additional resources necessary for its execution.
   b. Ensure that the repository is set to public access for review.
   c. Follow the development guide provided separately for specific details on structuring and presenting your repository.
2. **Google Slides Presentation Link [URL]:**
   a. Prepare your project presentation using Google Slides.
   b. Include your team code in the first slide of your presentation
   c. The presentation must be set to public access. Test opening the link in incognito mode to confirm its accessibility.
   d. **Only presentations in Google Slides format will be accepted.** Other formats like PDF or PPTX are **NOT** permissible.
   e. Ensure that your presentation is concise, visually engaging, and effectively communicates the key points of your project. It should be structured in a way that logically flows and is easy to follow.
3. **Documentation [PDF]:**
   a. Submit documentation of your project in PDF format.
   b. While there are no strict guidelines on the content, the documentation should comprehensively describe your project, adhering to general Execution and Expectation guidelines.
   c. This document is intended for reference purposes and future consultation, rather than assessment. It should include relevant information that offers clarity on your project's design, functionality, and other key aspects.

Note: Adherence to these guidelines is crucial. Failure to comply with the format requirements, especially regarding the Google Slides presentation, may result in penalties, including the possibility of disqualification. For further information and specifics, please consult the development guide mentioned below.

Below are the Judging Criteria that your project will be evaluated on:

Concept (weight: 40%)	Is the team targeting an area of necessity/need?	15%
	How accessible is the solution to the target audience?	15%
	To what extent does the team's solution consider user-friendliness?	10%
Execution (weight: 60%)	Is the solution sustainable as a business?	20%
	Does the team properly show that their solution is functional?	30%
	Does the team sufficiently argue that their solution is scalable?	10%

You will find in the Developmental Guide section that we highly encourage you to keep these criteria in mind during your development process.

8. Judging, Showcase, & Prizes

The competition consists of two judging rounds, with a showcase in between:

Round 1: Teams aligned with specific themes will compete against each other. Judges will select the top two teams to advance to the next round and identify one honorable mention team per theme. While the latter won't proceed, they will receive awards for their commendable efforts.

In each theme, each team will have exactly 4 minutes to present their solution, followed by 2 minutes of Q&A (further discussed under the Development Guide, and Pitch sections).

Showcase: Each team will have a booth in the walkway to present their work to a potential investor audience. Interested investors may express their desire to incubate and/or invest in the showcased projects.

Round 2: The six teams advancing to this round, two from each theme, will compete against each other in this final round. Judges will announce the top three winners on stage, with the remaining three teams designated as runners-up. The three winners do not have to be from different themes. All teams, including honorable mentions, will be acknowledged.

Prizes:
- 1st Place: 24,000 QAR *(per team)*
- 2nd Place: 16,000 QAR *(per team)*
- 3rd Place: 8,000 QAR *(per team)*
- Runner-Ups [3 teams]: 4,000 QAR *(per team)*
- Honorable Mentions [3 teams]: Gift and a 250 QAR gift card *(per member)*

9. Communications

Wifi Connection

Connect to the following WiFi network:

Name: **Tartan**
Password: **hackathon2024**

Discord

After receiving your acceptance email, you were required to join the LifeLines Hackathon '24 Discord server. This is where the bulk of your communication (between students, mentors, and organizers) will take place. Below are the most essential channels to you:

Channel Name	Channel Description
Announcements	- The Announcement channel will be the main communication channel between the event organizers and the participants. This is where meal calls, location directions, reminders, impromptu changes, important news, and all other official announcements will be made. - Only organizers will be able to post in the Announcements channel, so **make sure to keep the notifications on.** - Keep an eye out for messages from this channel to keep up with the event's progression.
Help Desk	#help-desk is where you can call for organizers help when needed to resolve issues that may arise. - You can tag @organizers or @Admin in this channel.
Mentor Help	- If you have a simple question, ask away and wait for a mentor to respond. - If you require in-person assistance, or if your inquiry can't be easily answered on Discord, call for a mentor to pass by your team (make sure to include your team's room and the type of support needed so that mentors will know how to best help you).
Introductions	- Introduce yourself and your role within this theme, and share your expertise or what excites you about this particular area. Connect with others selected for the same theme and kickstart your hackathon journey! - There is a general #introductions channel for all themes and each theme has a specialized #meet-me channel for members selected for a specific theme. If you do not have access to your theme's channel, then please ping @Admin in #help-desk to resolve your issue
Off-topic	A relaxed and casual space where you can chat about anything and everything not directly related to the hackathon. Whether it's sharing your latest hobby, a meme, or just having a fun conversation, this is your spot for lighthearted and diverse discussions!

Other Contacts

CMU Security: 4454 8550

Can be contacted for both security emergencies and to dispatch ambulances.

C. Development Guide

1. Ideation and Conceptualization

Understanding the Need

The initial phase of ideation and conceptualization is crucial in setting a strong foundation for your project. Begin by identifying and understanding the necessity or need your solution aims to address. This involves:

1. **Research and Analysis:**
    -   Conduct thorough research on the issue you plan to tackle. Look into existing solutions, if any, and analyze their effectiveness and shortcomings. There has to be a major degree of novelty in your solution.
    -   Understand the context and the extent of the problem. This might involve reviewing case studies, statistical data, or academic research.
    -   Engage with experts or those directly affected by the problem for first-hand insights.

2. **Gap Identification:**
    -   Based on your research, identify gaps in current solutions. Are there unmet needs or aspects that could be significantly improved?
    -   Consider innovative approaches or technology that hasn't been utilized effectively in this area.

3. **Documenting the Need:**
    -   Clearly document your findings and the specific need your project will address. This will serve as a guide throughout the development process.

Defining Accessibility

Your solution should be designed with the target audience in mind, ensuring it is accessible and practical for them. Focus on:

1. **Target Audience Analysis:**
    -   Define who your target audience is. Understand their characteristics, needs, and limitations.
    -   Research the resources and technologies available to your audience. This includes technology literacy, internet access, and other relevant factors.
2. **Accessibility Planning:**
    -   Develop a strategy to make your solution as accessible as possible to the target audience. This might involve using widely available technology, considering language barriers, and ensuring low-cost solutions.
3. **Validating Assumptions:**
    -   Regularly validate your assumptions about the target audience's needs and resources through user feedback, expert consultations, and testing.
    -   If you are making an assumption that you're not sure off, reach out to the mentors to make sure you can make that assumption or not
    -   Example: If your solution needs access to the internet, can you assume that your target audience has access to the internet during the occurrence of the catastrophe they're in? (war or natural disasters for example)

Focusing on User-Friendliness

The usability of your solution is paramount. It should be intuitive and easy to use for the intended audience.

1. **User Experience (UX) Design:**
    -   Design with the end-user in mind. The interface should be simple, intuitive, and devoid of unnecessary complexity.
    -   Consider the user journey and how users will interact with your solution from start to finish.

2. **Feedback Loops:**
    -   Incorporate mechanisms for receiving user feedback during and after the development process.
    -   Use this feedback to make iterative improvements, ensuring the solution remains user-friendly and effective.

2. Execution and Expectation

Sustainability and Business Viability

The Execution phase is where your concept materializes into a tangible solution. Key to this is ensuring sustainability and potential as a business or ongoing project. This involves:

1. **Business Plan Development:**
    -   Develop a comprehensive business plan that outlines how your solution will be sustained over time. Consider different models: for-profit, non-profit, or a hybrid.
    -   Address key components such as funding sources, revenue generation (if applicable), and cost management.

2. **Funding Strategies:**
    -   Identify potential funding sources. This could include grants, crowdfunding, venture capital, donations, or government funding, depending on your solution's nature.
    -   Provide realistic assessments and plans for securing these funds. Evidence of preliminary contacts or interest from potential funders can be advantageous.

3. **Long-term Vision:**
    -   Articulate a long-term vision for your solution. How will it evolve? How will ongoing costs be covered? What is the roadmap for the next 5-10 years?

Functional Prototype Development

While a fully working solution is not expected, a proof of concept or functional prototype is essential.

1. **Implementation Quality:**
   - Focus on creating a high-quality prototype that demonstrates the core functionality of your solution.
   - Ensure the prototype is stable enough to showcase the intended results without significant errors or issues.

2. **Code and Technical Documentation:**
   - Maintain well-documented and organized code. This not only proves the solution's functionality but also demonstrates professional technical execution.
   - Include technical documentation that explains how the solution works, its architecture, and any unique technical features.

Scalability

Demonstrating that your solution can grow and adapt to increased demand is crucial.

1. **Scalable Architecture:**
   - Design your solution with scalability in mind. This means using efficient algorithms, considering cloud-based solutions, and planning for increased loads.
   - Show an understanding of the technical requirements for scaling up, such as handling more users or more data.

2. **Resource Management:**
   - Plan for efficient use of resources. This includes optimizing code for performance, considering cost-effective infrastructure, and planning for resource allocation as the solution grows.

3. **Economic and Operational Scalability:**
   - Address how your solution can be economically scaled. This includes cost analysis for scaling up and strategies for managing increased operational demands.
   - Present a realistic and well-thought-out plan for how your solution can be expanded or adapted to larger or different user groups over time.

3. Documentation and Presentation

<u>Documentation</u>

Throughout the ideation and conceptualization phase, document your process meticulously. This documentation should include:

- The identified need and the research backing it.
- Analysis of the target audience and the accessibility strategy.
- The design decisions that are made to ensure user-friendliness.
- Changes made based on feedback and testing.

This documentation will not only guide your development process but also strengthen your presentation to the judges, demonstrating a well-thought-out and researched concept that addresses a genuine need in an accessible and user-friendly manner.

Throughout the execution, maintain comprehensive documentation of your development process, challenges faced, solutions implemented, and any iterations made. This documentation should include:

- Your business plan and funding strategy.
- Technical documentation of the prototype.
- Evidence and plans for scalability.

[Presentation (Pitch)](#)

Presenting this information effectively in your final pitch will demonstrate the thoroughness of your execution and the potential of your solution to be sustainable, functional, and scalable. Here is an example outline to help you get started:

- **Intro**
  - Introduce team members and their respective roles.
- **Problem Statement**
  - Define the area of need.
  - Highlight previous attempts to solve the problem (preliminary research).
  - Discuss the shortcomings of previous attempts.
- **Solution & Impact Overview**
  - Present your solution to the problem.
  - Explain the design and argue for functionality.
  - Describe how your solution makes a difference.
  - Quantify the impact to emphasize its magnitude.
- **Business Model**
  - Note that your solution doesn't have to be for-profit but must be financially sustainable.
  - Where would you get the funding? Argue that this avenue is achievable.
  - The business model can be used to sell your idea to the judges, so make it appealing.
- **Demo**
  - A video or screenshots of your proof of concept
- **Conclusion**
  - Summarize any key points, reinforcing the solution's impact.

Please note that these are by no means guidelines; you are free to structure your pitch however you see fit. However, keep in mind the judging criteria at all times.

Also note that:
- Each team is given **exactly 4 minutes** to make their pitch. This includes the demo time if the team wishes to have one. To stay on schedule, no additional time will be granted.
- Q&A time will be an additional **2 minutes** if the judges wish to ask questions. Teams may not use that time for their pitch, even if no questions arise.

4. Extra Tips

Brainstorm Tips

- Do background research first and foremost. You don't want to end up creating something that was already done.
- Consider how a new user would interact with your design; make sure it's intuitive and accessible to your target audience.
- Don't start your pitch with slides. It's easier to get your thoughts down on scratch paper first
- Always consider the judging criteria. Have them on hand and make sure you are covering your bases as you create your pitch.
- Start your pitch early, and don't wait for the code to define your solution. We strongly suggest that your team Specialists start the pitch as soon as the ideation and planning phases are done (i.e., at the same time that Hackers start the coding).

Presentation Tips

- Show interest and enthusiasm. Remember that you are essentially trying to sell an idea to the judges and audience.
- Utilize confident body language. You are trying to convince your audience, so you must appear sure and assertive.
- Don't exceed your allocated time. You will not be given an extra minute, and an incomplete presentation reflects badly on your project.
- Practice excessively; assign each presenter's slides and practice them together until you are confident in your presentation.

Q&A Tips

- Predict the judges' questions and try to answer them during your presentation. The judges will only have time for a few of their inquiries, and if something remains unclear to them it may be reflected in their judging. Again, refer to your evaluation criteria to predict possible questions.
- Prepare to have your assumptions questioned by the judges. You are responsible for convincing them of the sufficiency of your proofs & reasonings.

General Tips

- Don't spend too much time detailing the design; remember that you only need a proof of concept to sell your solution.
- Ask for help when needed! Mentors are there for that exact purpose.
- Take time to rest. Pushing yourself to an early burnout will take away from your enjoyment of the hackathon and will do your project no favors.

D. Technology and Tools

Throughout this hackathon, certain technologies will be used. See below any restrictions or suggestions:

- Code hosting:
    - You should create a GitHub repository and commit your progress there. **Your submission must include a link to your public GitHub repository.**
    - If you are not familiar with git and GitHub, you can refer to the internet: https://docs.github.com/en/repositories/creating-and-managing-repositories/quickstart-for-repositories
- Programming languages:
    - There is no restriction on the choice of the programming language or technologies used.
- Project deployment:
    - GitHub Pages allows you to host static web pages for free. https://pages.github.com/
    - Various Cloud Providers have free trials or free credits to use virtual machines. Here is a short list of benefits for around 10-15 providers: https://github.com/cloudcommunity/Cloud-Free-Tier-Comparison
    - Local deployments can be opened to the public using https://ngrok.com/, but be mindful of its security implications.
    - If you are planning to make a website using React, consider using Next.JS and free deployment via https://vercel.com
- Demo Video:
    - You could use Zoom, CamStudio, OBS Studio, Bandicam, and various other tools if you want to record a working Demo video for your pitch.
    - You can use iMovie (mac-exclusive), CapCut, or any other video editing software for quick editing work if needed.

E. FAQs

You can find the most common FAQs on our website.

If you have any other questions, please feel free to reach out in Discord.

Carnegie
Mellon
University
Qatar
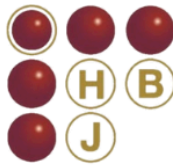
مركز حمد بن جاسم
لتعليم علوم الحوسبة
HAMAD BIN JASSIM CENTER
FOR K-12 COMPUTER SCIENCE EDUCATION

التعليم education
فوق above
الجميع all

الهلال الأحمر القطري
Qatar Red Crescent

avey

رابطة سيدات الأعمال القطريات
QATARI BUSINESSWOMEN ASSOCIATION

QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

+974 5517 0084
lifelines@qatar.cmu.edu
www.lifelinescmuq.org