

VEERSA HACKATHON – DCRUST BATCH OF 2026

**COME JOIN US AND SHOW YOUR EXPERTISE AND PASSION IN SOLVING SOME
INTRIGUING CHALLENGES IN THE HEALTHCARE INDUSTRY**

HACKATHON GUIDELINES

1. This hackathon is exclusively for DCRUST 2026 batch students who have been invited to participate in the Veersa selection process via an e-mail.
2. Form a team of Minimum of 3 and Maximum of 4 members and give a relevant name for your team.
3. Register for the hackathon at <https://forms.office.com/r/XPmVMfMYCA>, no later than **7:00 PM IST on December 24, 2025 (Wednesday)**.
4. Please submit the Form only once per team. Multiple entries will lead to disqualification.
5. Deadline to submit the project is **11:59 PM IST on December 27, 2025 (Saturday) on https://forms.office.com/r/RU7Cj5YNz0**.
6. Please strictly adhere to the below guidelines for your submissions.

TECHNICAL GUIDELINES

1. The solution with an interface must include an Information Architecture and actual Figma/ Adobe XD or another design file
2. You can use any language to build your solution.
3. You can use any open-source tool, or the tools mentioned in the list below to build your solution
4. Your code must be modular and reusable across components

QA GUIDELINES

1. Participants need to deliver the best possible solution within the timeline given. They need to detect any issues that might block the successful operation or affect the user experience.
2. To achieve that, they need to document the requirements to solve the problem statement.
3. Against each requirement, they need to write valid test cases and test the software.
4. For testing, they can follow any 2 approaches:
 - a. Manual Test Cases in a document
 - b. Automated Unit Test cases: where the developers write unit tests for checking the reliability of their written code.
 - c. Automated API Test Cases: It is similar to the unit test cases with a setup, a call to the API method, and then checking the results.

- d. Record and Play: Can use any open-source tool to record and play, and check the reliability of the software.
 - e. UI Automation: Creation of automated scripts to check the successful operation of the software.
5. The participants can use any source tool or any language for automation.

CODE REPOSITORY GUIDELINES

1. Participants must use Github for managing their codebase
2. Participants should make sure that the repository that they are sharing with us is public.
3. Participants should not make only a single commit/push to repository with the final codebase, they should start with an empty repository and then keep on committing and pushing their code at some checkpoints of their choice.
4. Individual has to check-in the code using their own id and not use a generic id.

HOSTING/DEPLOYMENT GUIDELINES

1. Participants can choose wherever they want to host and deploy their application.
2. Some of free to use service providers are Heroku, Netlify, Firebase, AWS etc.

SUBMISSION GUIDELINES

1. Label your Project as per your team's name
2. The working prototype must be hosted and link published on the Github repository
3. The cleaned and well labelled code must be present in the same Github repository
4. Figma/ Adobe XD links, Documented/ automated test cases etc. to be uploaded on the Github repository
5. A short video summing up your presentation in no more than 5 minutes uploaded on the Github repository itself

The link to your Github repository completed in all respect must be submitted on MS Forms- <https://forms.office.com/r/RU7Ci5YNz0> no later than 11:59 PM IST on December 27, 2025 (Saturday).

SELECTION CRITERIA

- Technical Solution – Everyone must present their work.
 1. How technically impressive was the hack
 2. Was the technical problem the team tackled difficult
 3. Did it use a particularly clever technique
 4. Did it use many different components
- Design pattern –
 1. Did the team put thought into the user experience
 2. How well designed is the interface

- Quality Check – Did the team followed all quality parameters and did performance testing.
- Completion –
 1. Does the hack work
 2. Did the team achieve everything they wanted
- Learning –
 1. Did the team stretch themselves
 2. Did they try to learn something new
 3. What kind of projects have they worked on before

TOOLSETS WHICH CAN BE USED

1. Daily Web RTC Video/ Audio component - [Sign up | Daily](#)
2. Square Up payment gateway - [Developer APIs: Payments, Orders, Customers - Square \(squareup.com\)](#)
3. DeepGram transcription service - [Create an account \(deepgram.com\)](#)
4. Google location and distance API - <https://cloud.google.com/blog/products/maps-platform/how-calculate-distances-map-maps-javascript-api>
5. Google API for Natural Language processing - <https://cloud.google.com/natural-language>
6. Apple chat API - <https://register.apple.com/resources/messages/msp-rest-api/>
7. AWS Medical comprehend - <https://aws.amazon.com/comprehend/medical/>
8. Open AI whisper - <https://openai.com/index/whisper/>
9. Adverse events repository - <https://open.fda.gov/data/faers/>

REQUIREMENTS

Use case 1 – Adverse medical event prediction from a phone call

Problem Statement – Millions of conversations happen between medical agents/ nurses and patients everyday regarding their medical conditions and medications. While the patients or nurses might not realize, there might be indications of an adverse event which has already occurred or there is a possibility of one occurring. It is critical to identify and flag such possible adverse events to quickly and accurately to prevent the same from occurring or remediating the one that has happened already.

1. Please develop a solution with the end objective to identify and predict any possible adverse event from a recorded conversation between patient and nurse/ medical agent
2. You can use the data available in the adverse events repository shared in the tool library above to develop your prediction model

3. You can create a conversation or use any publicly available conversation to demonstrate your solution
4. Before actual prediction, detailed data analysis is expected to support the model.

Use case 2 – Telehealth Solution for access to Healthcare from anywhere.

Problem Statement – Post pandemic, world has changed significantly. Quick access to quality, affordable and reliable healthcare from anywhere is the need of the hour. People are confined in remote locations in their homes and hence there is an urgent need to come up with a digitally enabled solution to address the following –

1. Instant access to Health consultation similar to in person experience by developing a Telehealth application
2. Capture basic patient information and particular specialty for which Telehealth consultation is needed
3. Ability to accept payment for the Telehealth consultation before the actual consultation begins
4. No compromise in the privacy of the consultation involving PHI data and the ability to securely save the information for future reference
5. Ability to chat and provide information from within the application during Telehealth session
6. Overcome challenges of difference in dialect/ accent in remote consultation via a transcription service

Use case 3 – Mobile application for scheduling of an appointment for a patient with doctor.

Problem Statement – Doctors are lifesavers and that is why they shouldn't be too far from those who need healthcare. With more and more people having the need to seek medical attention, doctors need to make their services accessible and available to their patients. In this time and age where everything and anything can be found on the internet, we need to take advantage of the digital tools/ channels to enable quick scheduling of appointments. Please envisage and develop a digital solution to give patients the freedom to book the appointments at any time and from anywhere including the following features –

1. Find availability of all doctors basis their specialty
2. Ability to find doctors near your location with doctors list sorted basis the shortest distance
3. Book the appointment and receive confirmation notification in the app.

4. There shall not be any conflicting consultation for the doctors/patients at the booked time slots.
5. Reminder notification one hour before the appointment time with a link to driving directions to doctor's clinic

Use case 4 – Readmission prediction for heart failure patients. (AI/ML)

Problem Statement - Heart failure is a very common ailment leading to fatalities if not attended to promptly. Even for the patients who get proper treatment, hospital readmissions result in a significant risk of death and a financial burden for patients, their families as well as the already overburdened healthcare systems. Prediction of at-risk patients for readmission allows for targeted interventions that reduce morbidity and mortality.

1. Develop a machine learning model with the end objective to predict readmission of heart-failure patients within 30-days of discharge from the hospital.
2. We have provided a subset of tables at the below one-Drive link for this problem.

[Veersa Hackathon Submission](#)

3. Participants will further decide which tables they will use to solve the problem.
4. Link to mimic-III table descriptions. - <https://mimic.mit.edu/docs/iii/tables/>
5. Before actual prediction, detailed data analysis is expected to support the model.
6. Below are the list of Diagnosis codes (Icd9_codes) representing heart-failure:
('39891','40201','40211','40291','40401','40403','40411','40413','40491','40493','4280','4281','42820','42821','42822','42823','42830','42831','42832','42833','42840','42841','42842','42843','4289')