



IIT KHARAGPUR

# INTER IIT TECH MEET 10.0

25-27TH MARCH 2022

## MERCARI'S LARGE SCALE SYSTEM DESIGN HACKATHON

Mercari is a community-powered C2C marketplace app from Japan, allowing anyone to buy and sell anything from anywhere. The primary objective here is to develop geographically scalable and accessible solutions. Thus, as an exercise of similar situations that we face daily, we propose to foster a Hospital Management System (HMS) [1] solution accessible on a large scale through this problem statement. HMS is inspired by the National [Digital] Health Mission (NHM) [2] of providing a digital healthcare footprint to every Indian citizen through the National Health ID (NHID). It aims to solve the complications coming from managing all the paper works - handwritten prescriptions, test reports, hospital/specialist referrals, etc., of every patient manually.



| LOW PREP

# INTRODUCTION

A Hospital Management System (HMS) is a software application that helps manage the information related to health care and track patients through their treatment journey. On the scale of a country as large and diverse as India, the problem requires innovative, scalable solutions. Creating a network of approved hospitals, HMS will provide the ability to manage all the records in one place, visualize a patient's medical history, allow doctors to manage patients better by automating manual workflows (example, automating the generation of medical bills and preparing the medicines in the hospital's dispensary based on doctor's appointment conclusion).

## PROBLEM STATEMENT

Develop a mobile-based, highly scalable solution [19,21] for Multi-Chain Hospitals under NHM. The solution should cater to patients and doctors to help them with observation, diagnosis, treatment, appointment scheduling, referrals, record visualization, etc.

You are free to use your creativity with the platform, technologies, features, etc. The minimum basic guidelines for the HMS are:

- Every citizen has NHID linked with them
- Hospitals (and their doctors) are associated with NHM
- Usage of Microservice architecture [3,19,22] is highly recommended
- Digital storage [4] and linking of data [5] (i.e., bills, reports, prescriptions, hospital referrals, etc.) with the proper owner
- Usage of Docker [6], Kubernetes [7], Travis CI [8], GitHub Actions [9], Spinnaker [10], and other open-source technologies is encouraged for continuous deployment and continuous integration purposes
- Usage of Datadog [11], Sentry [12], and other free (or trialware) for application performance and error monitoring, load balancing, etc.
- Following proper code and documentation etiquettes [13, 14], architecture practices [15], and production security and development operations [16] are encouraged
- Considering maintainability, writing unit and integration tests [17] are recommended
- The use of existing open-source (or trialware) projects, APIs, and technologies are highly encouraged as felt appropriate. Using tools provided in the GitHub Student Pack is an excellent way to incorporate paid solutions if required.

It is VERY IMPORTANT to design your architecture because scalability is the primary issue in a large country like ours; whether it's booking tickets on IRCTC or checking university results, we have all faced it. [18]

\* Number in bracket refers to the reference number.



| LOW PREP

# SUBMISSION REQUIREMENT

- GitHub (or any other code hosting platform) repositories collection of Microservices (30 points)
- Deployed fully-functional instance(s) of the designed service(s) (80 points)
  - Link(s) to the deployed microservices
  - Demo video of the solution demonstrating the running of the application and system
- A brief report, not exceeding 6 pages (excluding cover page, index, etc., i.e., content only), demonstrating the design decisions (40 points)

The deadline for all the submissions is 24th March, 11:59 p.m.

**Send your submissions at this email: [submissions@interiit-tech.org](mailto:submissions@interiit-tech.org)**

# EVALUATION

The evaluation focuses on visualization and usability of the interface(s), scalability of the designed architecture, and maintainability of the submitted project repositories. The decisions you make while designing the architecture and planning how to maintain (and develop further) the produced code is critical as thousands of other developers would be working alongside you.

A specific non-exhaustive list of evaluation criteria is provided below.

- Qualitative observation of Patient, Doctor, and NHM Admin features
- Design and Usability of the design interfaces
- Ease of CRUD operations over data by the respective owner(s) and authority
- Scalability of the Architecture & System Design
- Ease of deployment and development, emphasizing a large team of remote engineers (i.e., implementation of CI/CD, modular code design, etc.)
- Technology decisions affecting network latency, deployment time, etc. For example, Inter-microservice communication decisions among REST, GraphQL, gRPC, etc.
- Monitoring, handling, and notifying about failures (partial or complete), availability, errors, etc.

Dataset: [Dataset#](#)

# You are free to use other open-source free-license data and images as required.



| LOW PREP

# REFERENCES

1. <https://www.leadsquared.com/what-is-hospital-management-system/>
2. [https://en.wikipedia.org/wiki/National\\_Digital\\_Health\\_Mission](https://en.wikipedia.org/wiki/National_Digital_Health_Mission)
3. <https://towardsdatascience.com/microservice-architecture-and-its-10-most-important-design-patterns-824952d7fa41>
4. <https://aws.amazon.com/blogs/architecture/store-protect-optimize-your-healthcare-data-with-aws/>
5. <https://cloud.google.com/storage/docs/encryption>
6. <https://aws.amazon.com/docker/>
7. <https://kubernetes.io/>
8. <https://www.katalon.com/resources-center/blog/benefits-continuous-integration-delivery/>
9. <https://github.com/features/actions>
10. <https://spinnaker.io/>
11. <https://www.datadoghq.com/>
12. <https://sentry.io/>
13. <https://www.techopedia.com/definition/16626/error-handling>
14. <https://bigsea.co/ideas/coding-étiquette-6-obvious-overlooked-best-practices/>
15. <https://techbeacon.com/app-dev-testing/top-5-software-architecture-patterns-how-make-right-choice>
16. <https://www.ibm.com/cloud/learn/devsecops>
17. <https://engineering.linkedin.com/blog/2016/08/writing-maintainable-integration-tests>
18. [https://charlieinden.github.io/System-Design/2021-01-25\\_Microservices--Designing-Highly-Scalable-Systems-55dbb6f64c94.html](https://charlieinden.github.io/System-Design/2021-01-25_Microservices--Designing-Highly-Scalable-Systems-55dbb6f64c94.html)
19. <https://www.fullstackpython.com/microservices.html>
20. <http://highscalability.com/start-here/>
21. <https://github.com/mercari/mercari-microservices-example>
22. <https://engineering.mercari.com/en/blog/tag/microservices/>

## Images/Data References used in the dataset:

1. <https://github.com/tsg-iitkgp/inter-iit-tech>

Team size for this event is maximum 6 participants.  
Participation awards shall be awarded to all participants.



| LOW PREP