Mastersolis Infotech – 22 Hour AI Hackathon

Organized by Mastersolis Infotech

Introduction

Welcome to the **22-hour Hackathon** organized by **Mastersolis Infotech**.
This event is designed to test your ability to create **AI-powered full-stack software solutions** that address real-world business challenges.
Participants will collaborate in teams, use AI tools creatively, and showcase their technical excellence through innovative project development.

General Hackathon Guidelines

1. The hackathon duration is **22 hours**. Teams must plan and manage their time efficiently.
2. Each team will consist of **4 members**. Role distribution among frontend, backend, AI integration, and design is encouraged.
3. Participants must choose **only one** problem statement from the two provided.
4. All development should be done **locally**. Cloud deployment (AWS, Azure, etc.) is **not required**.
5. Each team must **push the final project** to a **public GitHub repository** before the submission deadline.
6. Participants are encouraged to use **AI tools** such as ChatGPT, GitHub Copilot, AWS Bedrock, Gemini, Claude, or Hugging Face for:
    - Code generation and debugging
    - UI/UX design assistance
    - Documentation and content writing
    - Testing and automation support
7. Internet usage is permitted **only for development-related work**.
8. Teams must **maintain silence and discipline** inside the venue.
9. **Doubts or clarifications** will be addressed only during mentor visits. Company representatives will visit tables periodically to answer queries.
10. Teams must **save and back up their work** regularly, both locally and on GitHub.
11. **Plagiarism is strictly prohibited.** Reusing prebuilt project templates or existing codebases will result in disqualification.

12. Evaluation will be based on **innovation, functionality, AI integration, design quality, and presentation**.

Hackathon Schedule (22 Hours)

Phase	Duration	Description
Phase 1	0 – 1 hour	Problem briefing, setup, and team planning
Phase 2	1 – 15 hours	Core development and AI integration
Phase 3	15 – 18 hours	Testing, debugging, and improvements
Phase 4	18 – 21 hours	Preparing final demo and GitHub submission
Phase 5	21 – 22 hours	Final presentations and evaluation

Judging Criteria

Criteria	Weightage
Innovation & Use of AI	25%

Technical Implementation & Code Quality	25%
UI/UX Design and User Flow	20%
Completeness of Features	20%
Presentation & Clarity	10%

Problem Statement Options

Teams must choose **one** of the following problem statements.
Each represents a practical scenario relevant to Mastersolis Infotech's business direction.
Participants are expected to apply **AI tools creatively** across different stages of development.

Problem Statement 1: AI-Powered Company Website Builder

Objective

Build a modern, dynamic, and responsive **company website for Mastersolis Infotech** using AI-assisted tools.
The website should showcase the company's services, achievements, and culture, while integrating automation and intelligent features.

Core Pages and Features

1. Home Page

- Hero banner, tagline, and dynamic content (AI-generated).
- Testimonials and service highlights.

2. About Page

- Mission, vision, and values.
- AI-generated team introductions and company milestones.

3. Services Page

- Service listing with AI-generated descriptions.
- Optional AI chatbot for visitor engagement.

4. Projects Page

- Portfolio of past projects with tag-based search and filtering.

5. Contact Page

- Contact form storing user submissions in the database.
- Automatic email responses using AI-generated messages.

6. Admin Dashboard

- Admin login to manage site content.
- Analytics dashboard with AI-based summaries.

1. Functional Enhancements

a. Career Page

- A dynamic job-listing section where the admin can post open positions.
- Candidates can apply through a form that stores applications in the database.
- Automatic email acknowledgment using AI-generated text ("Thank you for applying to Mastersolis…").

b. Blog / News Page

- Admin can add blog posts or company updates.
- AI can generate blog summaries or SEO descriptions.
- Optional "AI-summarize" button for visitors to get short summaries of long posts.

c. Testimonials & Case Studies

- A section that uses AI to generate or rephrase testimonials based on client data.
- Case study upload with AI-assisted summary generation.

D.**Resume Filtering tool** for all the applications received on careers page for admin to filter out applications

The tool should automatically **extract candidate information** (like skills, experience, and education) and **format it into professional templates**, ensuring uniformity for recruiters. It can also include features such as **AI-based resume suggestions**, **job-fit analysis**, and **ATS-friendly formatting** to improve both candidate and recruiter experience.

⚙️ Key Features:

- User-friendly **resume creation interface** (drag-and-drop or form-based)
- **Predefined professional templates** with customization options
- **Smart suggestions** for skills and achievements based on job role
- Option to **download or directly submit** the resume for selected job openings
- Integration with company's **careers page** and backend **applicant tracking system (ATS)**
- (Bonus) **AI resume scoring** or **job match percentage** feature

🧠 Expected Outcome:

A functional web module that helps candidates easily build and apply with a polished resume — making the company's recruitment process smoother, faster, and more engaging.

Suggested Tech Stack

Area	Suggested Tools
Frontend	React
Backend	Flask / Django
Database	PostgreSQL

AI Tools	ChatGPT / AWS Bedrock / Gemini / Hugging Face
UI/UX Design	Figma

Problem Statement 2: College Placement Management Portal

Objective

Develop a **Placement Management Portal** to streamline campus recruitment processes for colleges.
The system should support multiple roles — **Student, Head of Department (HOD), and Training & Placement Officer (TPO)** — and manage all aspects of placement activities.

User Roles and Responsibilities

1. Placement Officer (TPO / Admin):

- Create and manage placement drives.
- Add company details (name, role, eligibility, etc.).
- Manage recruitment rounds and update selected/rejected lists.
- Upload offer letters for selected students.
- Send email notifications at each stage.

2. Head of Department (HOD):

- Approve student signups within their department.
- Verify and edit student profiles.
- View department-level placement statistics and performance reports.
- Generate downloadable reports (Excel/PDF).

3. Student:

- Sign up and log in securely.
- Maintain personal details and upload résumé.
- View and enroll in active drives.
- Track drive status and round results.
- Receive automated emails for selection or rejection.
- Download final offer letters.

Core Features

- Role-based authentication and authorization.
- Placement drive creation and management.
- Student registration and résumé upload.
- Automated email notifications.
- Department-wise, offer-wise, and student-wise report generation.
- Dashboard for analytics and insights.

AI Integration Opportunities

- AI-generated email templates and content.
- AI-assisted résumé parsing and validation.
- Automated summarization of placement results and insights

Resume Filtering tool for all the applications received on careers page for admin to filter out applications

The tool should automatically **extract candidate information** (like skills, experience, and education) and **format it into professional templates**, ensuring uniformity for recruiters. It can also include features such as **AI-based resume suggestions**, **job-fit analysis**, and **ATS-friendly formatting** to improve both candidate and recruiter experience.

⚙️ Key Features:

- User-friendly **resume creation interface** (drag-and-drop or form-based)
- **Predefined professional templates** with customization options
- **Smart suggestions** for skills and achievements based on job role
- Option to **download or directly submit** the resume for selected job openings
- Integration with company's **careers page** and backend **applicant tracking system (ATS)**
- (Bonus) **AI resume scoring** or **job match percentage** feature

🧠 Expected Outcome:

A functional web module that helps candidates easily build and apply with a polished resume — making the company's recruitment process smoother, faster, and more engaging.

Suggested Tech Stack

Area	Suggested Tools
Frontend	React
Backend	Flask / Django
Database	PostgreSQL
AI Tools	ChatGPT / Bedrock / Gemini / Hugging Face
UI/UX Design	Figma

Problem Statement 3:

AI-Based Timetable Generation System aligned with NEP 2020 for Multidisciplinary Education Structures

Description

With the implementation of the National Education Policy (NEP) 2020, all higher education institutions-including Institutes of Teacher Education and general degree colleges-have transitioned to Four-Year Undergraduate Programmes (FYUP) and integrated teacher education programs like B.Ed., M.Ed., and ITEP. These programs feature flexible, creditbased, multidisciplinary structures allowing students to choose Major, Minor, Skill-Based, Ability Enhancement, and Value-Added courses. Manual timetable creation under this new framework has become extremely complex, involving numerous subject combinations, varying credit hours, student preferences, and faculty workload distribution. To manage

this, colleges usually form dedicated timetable committees, but even then, clashes, underutilization of faculty, and scheduling inefficiencies persist.

This problem statement envisions the development of a web-based or hybrid intelligent system that can generate automated, conflict-free, optimized academic timetables alignment with the NEP 2020 course structure. The system must integrate:

- Student data (elective choices, enrolled credits)

- Curriculum structure (course codes, credits, theory/practical split)

- Faculty workload, availability, and expertise

- Room/lab availability and capacity

- Teaching practice schedules (especially relevant for B.Ed. and M.Ed.)

- Field work, internships, and project components

The system should also allow dynamic editing, scenario simulation, and scalability for upcoming semesters. A user-friendly admin interface is essential to allow real-time inputs and updates.

Expected Solution:

A deployable AI/ML-assisted Timetable Generator that:

- Accepts structured inputs from academic, student, and faculty databases

- Generates a semester-wise timetable for multiple programs (B.Ed., M.Ed., FYUP, ITEP)

- Prevents scheduling conflicts across faculty and infrastructure

- Accommodates future course additions and changing NEP guidelines

- Offers exportable formats (PDF, Excel) for student and faculty sharing

- Can be integrated with existing Academic Management Systems

---

Submission Instructions

1. All development must be completed during the hackathon period.
2. Teams must push the **final project** to a **public GitHub repository** before submission.
3. The repository should include a **README file** with:
    - Project title and chosen problem statement
    - Setup and run instructions
    - Team members and roles
    - Description of AI tools used
4. Ensure your project runs properly when cloned on another system.
5. Teams must present their project live during evaluation.

🧩 Doubt Clarification Process

During the 22-hour hackathon, over 150 teams will be working on complex AI-integrated software projects.
To ensure that every team gets fair and efficient guidance, **Mastersolis Infotech** has established a structured **Doubt Clarification System** using a Google Form.

This system helps mentors track and resolve doubts quickly while maintaining discipline and focus in the hall.

Guidelines for Raising Doubts

- Doubts will be addressed **only during scheduled mentor visits** or **doubt-solving rounds**.
- Participants should **not call mentors directly** unless explicitly instructed.

- Each team must use the **official Google Form** to submit any doubts.
- Teams should try to **use AI tools (ChatGPT, Copilot, Bedrock, Gemini, etc.)** before submitting a doubt.
- The form should include a **clear, concise description** of the issue and what was already tried.
- Common or repeated issues may be addressed **to all teams at once** via announcement.
- All teams must **maintain silence** and allow mentors to move between tables efficiently.

Doubt Submission Form

Use the following form to submit your doubts at any point during the hackathon:

👉 **[Google Form Link – https://forms.gle/nsqDgAHru4zN67rK9**

You can also scan the QR code displayed in the venue to access the form instantly.

Form Fields Overview

Each submission should include the following information:

Field	Description
Team Name	Enter your registered team name.
Table Number / Team ID	Helps mentors locate you quickly.
Problem Type	Choose from: Frontend, Backend, Database, AI Integration, GitHub, or Other.
Doubt Description	Summarize the issue in 1–2 lines.

What You Tried So Far	Mention steps or AI prompts already used.
Urgency Level	Choose High / Medium / Low.

Doubt-Solving Schedule

Session	Time	Activity
Round 1	3:00 PM – 5:00 PM	First round of mentor visits and doubt clarifications
Round 2	7:00 PM – 10:00 PM	Night round for ongoing development issues
Round 3	8:00 AM – 10:00 AM (next day)	Final round for wrap-up and final touch-ups

Note: Schedule may slightly vary depending on progress.

Rules for Interaction

- Mentors will spend a **maximum of 5 minutes per team per visit**.
- If your doubt requires deeper discussion, it may be noted for the **next round**.
- Mentors will **not debug line-by-line**, but will **guide approaches and unblock issues**.
- Teams are encouraged to **note doubts in a shared list** before filling the form.
- Please **save your project and push updates to GitHub frequently** to avoid data loss.

Tips for Effective Doubt Resolution

1. **Be specific:** Instead of “My backend isn’t working,” say “Flask API not returning data from PostgreSQL.”
2. **Use AI smartly:** Try at least one or two AI-based solutions first and note what you got.
3. **Stay calm:** It’s fine if things break — debugging under pressure is part of the hackathon learning.
4. **Listen to announcements:** Some issues may be explained to everyone at once.
5. **Respect time:** Let mentors assist all teams fairly.

Mentor Notes

Mentors will use the Google Sheet linked to the form to:

- Track all doubts by team and urgency.
- Sort and prioritize based on problem type.
- Record which teams were visited and resolved.
- Identify common patterns to announce quick collective solutions.

Summary

- ✅ Submit doubts **only via the official Google Form**.
- 🧠 Use **AI tools first**, then seek mentor guidance.
- 🔕 Maintain **silence and focus** during active work hours.
- 🕒 Mentors visit tables **only during scheduled rounds**.
- 💡 Be concise, specific, and patient — help us help you efficiently!

Final Note from Mastersolis Infotech

This hackathon encourages you to **think creatively, code intelligently, and collaborate effectively**.
Leverage AI tools not only for coding but also for documentation, UI/UX improvement, and innovation.
Deliver a project that reflects your team’s creativity and technical skill — something that could one day power a real Mastersolis Infotech solution.
