

The background features a large, dark blue curved shape on the left side. To its right is a bright yellow curved shape. In the bottom-left corner, there is a teal-colored shape with a fine, repeating geometric pattern.

Designing your Hackathon Project

Figuring Out a Hackathon Project

Coming
Up With
An Idea

Planning
Your
Approach

Executing
your Ideas

Testing
and De-
Bugging

Preparing
your
presentation

Coming Up With An Idea

Good hackathon projects start with a strong idea that is interesting and innovative, but takes into consideration the limited time-frame & resources.

1. Brainstorming

- All ideas are welcome at this stage!
- Let everyone contribute their thoughts
- Use something you can all write on (digital or analog)
- You want as many ideas as possible

2. Narrow it down

- Go through the list together and see if which ideas stand out to you.
- Are there themes / areas that repeat?
- This will help you figure out the problem you want to solve
- Think of real-world problems to guide you

3. Focus on the problem

- Hone in on your problem
- Why is it an interesting, relevant, or important problem?
- Which of your ideas from step 2 solve that problem?
- Who will the solution be for?

4. Define your idea

- How does your idea solve the problem?
- What resources/knowledge do you need?
- Who is on your team? What skills do they have?
- Can you do it in a weekend?
- Prioritize feasibility over complexity!
- You might have to discard a couple of ideas to find your final one.

Planning Your Approach

Set goals

What are your objectives? What do you want your project to be able to do?

Define your MVP (minimum viable product)

The most basic version of your project that meets your main goal
Include only the essential features needed.

Outline your project and set milestones

Figure out the order in which things need to get done and how long it will take. Over-estimate! This will help you prioritize and delegate.

Figure out what skills you need and who has them

Use your team wisely. Figure out who is best prepared to handle each part. Figure out how and when you will ask each other for help

Executing your Idea

Start with the basics

Start with the essential features of your project. Focus on basic functionality. You can add extra cool features later.

Be ready to iterate quickly

Things never go fully according to plan. If you can't make something work, don't get stuck! Come up with an alternative.

Test as you go along

Make sure you're testing your code frequently. It's easier to fix problems if you catch them early!

Collaborate and communicate

Check in with each other about your progress. Make sure everyone is on the same page and work out ways to help each other when you get stuck.

Testing and De-Bugging

Test your prototype internally. Review each other's sections and make a list of issues that need to be fixed, prioritized by how critical they are

Run your project by someone who doesn't know it as well as you do. Ask one of the mentors or another team to try it out.

Fix critical issues first! Check to see if something you fixed broke something else.

Focus on final touches. Polish the UI and make sure the prototype runs smoothly on different devices. You can add small "extras" but make sure they don't break anything!