# PREREQUISITES AND BEST PRACTICES FOR THE HACKATHON

Before starting the hackathon, it's important to make sure you're fully prepared and have all the necessary tools and knowledge to successfully complete the milestones. Below are some key prerequisites and best practices to help you stay organized and productive during the hackathon.

Event Date: 6th September 2024
Duration: 24 Hours (Starting Midnight after the event)
Submission Deadline: 7th September 2024, Midnight

Ameen Alam
Dean of Faculty

## Table of Contents

## Prerequisites:

1. **Basic Knowledge of HTML, CSS, and TypeScript:**
    - Ensure you are comfortable with creating a basic webpage layout using HTML.
    - Understand CSS for styling elements and making your page responsive.
    - Familiarize yourself with TypeScript basics, such as variables, functions, and DOM manipulation.
2. **Set Up Your Development Environment:**
    - Install a code editor like **VS Code**.
    - Make sure you have **Node.js** and **npm** installed, which are required to work with TypeScript and other dependencies.
    - Set up **Git** on your machine for version control and ensure your GitHub account is ready.
3. **Vercel Account:**
    - Create a **Vercel** account (if you don't have one already) and ensure you know how to deploy a project to Vercel from GitHub.
4. **Understand Project Structure:**
    - Be familiar with how to structure your project files (e.g., creating separate folders for HTML, CSS, and TypeScript files).
    - Know how to link CSS and TypeScript files with your HTML page.

## Best Practices During the Hackathon:

1. **Plan Before You Code:**
    - Before diving into the code, plan out your resume layout. Sketch how the sections (Personal Info, Education, Skills, etc.) will look and be positioned on the page.
    - Think about how you'll break down the project into smaller, manageable tasks to complete each milestone.
2. **Keep Your Code Clean and Organized:**
    - Follow proper indentation and naming conventions. This will help you (and others) understand your code later.
    - Use **meaningful variable names** in TypeScript, and make sure your CSS classes are descriptive (e.g., .resume-header instead of .header1).
3. **Test Regularly:**

- Don’t wait until the end to test your project. After completing each feature, test it in the browser to ensure it’s working as expected.
- Test for responsiveness by resizing your browser window to see how your design looks on different screen sizes.

4. **Use Git for Version Control:**
   - Commit your code regularly. This will allow you to track changes and roll back to a previous version if something goes wrong.
   - Push your code to **GitHub** frequently to avoid losing progress.
5. **Optimize for Performance:**
   - Make sure your CSS and TypeScript files are not bloated. Only include styles and code that are necessary.
   - Minimize the use of external libraries unless absolutely needed, as they can slow down the page.
6. **Make Use of Online Resources:**
   - If you get stuck on a particular issue, feel free to use online resources like **MDN Web Docs**, **W3Schools**, or **Stack Overflow**. However, avoid copying large chunks of code without understanding it.
7. **Time Management:**
   - Keep track of time and avoid spending too long on one small task. If something is taking longer than expected, try to move on and come back to it later.
   - Break your 24-hour hackathon time into manageable chunks and take breaks to avoid burnout.
8. **Collaborate and Communicate:**
   - Although this hackathon is an individual task, don’t hesitate to communicate with your peers or instructors if you need help.
   - If you encounter technical issues deploying on Vercel or GitHub, reach out early to resolve them.
9. **Backup Your Work:**
   - Always keep a backup of your project files, either in a cloud storage service or locally on your machine. This ensures you don’t lose progress due to technical issues.

## Additional Tips:

- **Read the Instructions Carefully:** Make sure you understand the requirements of each milestone before starting. It’s easy to miss details when you’re in a hurry, so take the time to read the instructions thoroughly.
- **Keep it Simple:** Start with the basics and gradually add complexity. Don’t aim for perfection on the first try; instead, focus on building a working version first.
- **Focus on Learning:** While prizes are motivating, the real goal is to challenge yourself and learn new skills. Don’t worry if you can’t complete all milestones—do your best and enjoy the process.

By following these prerequisites and best practices, you will set yourself up for a productive and successful hackathon experience. Good luck and remember to have fun while coding!