**2. Planning Phase**

**Objective**

The goal of the planning phase is to define a clear roadmap for building a simple, static/dynamic food ordering website using HTML, CSS, and basic JavaScript. This plan ensures that development is structured, timely, and meets the basic requirements for a functional UI-based ordering system.

**Project Scope**

**In Scope:**

* Design of a multi-page food ordering website (e.g., Home, Menu, Contact)
* Static content using HTML and CSS
* Basic interactivity using JavaScript:
  + Add to cart feature
  + Item count and price calculation
  + Order confirmation via alert/popup
* Responsive design using media queries
* Hosting the project on GitHub Pages

**Out of Scope:**

* User authentication/login
* Real-time database or backend server
* Payment gateway integration
* Admin dashboard

**Timeline / Milestones**

| **Week** | **Task** | **Milestone** |
| --- | --- | --- |
| 1 | Requirement analysis, wireframe sketches | Finalized layout design |
| 2 | Build HTML structure (home, menu, cart pages) | Basic site layout completed |
| 3 | CSS styling and responsive design | Responsive UI ready |
| 4 | Implement JavaScript for cart and interactivity | Cart system functional |
| 5 | Testing and fixing bugs | All features tested |
| 6 | Hosting and final documentation | Site deployed and documented |

**Tools and Technologies**

| **Type** | **Tools/Technologies** |
| --- | --- |
| Code Editor | Visual Studio Code |
| Languages | HTML, CSS, JavaScript |
| Design | Figma / Sketch / Pen & paper (wireframes) |
| Hosting | GitHub Pages / Netlify |
| Version Control | Git + GitHub |
| Testing | Manual browser testing (Chrome, Firefox, mobile) |

**Resource Plan**

Since this is a solo project (or student project), the roles and responsibilities are simplified:

| **Role** | **Responsibility** |
| --- | --- |
| Developer (You) | Design, code, test, and deploy the website |
| Reviewer (optional) | Classmate or teacher for feedback/testing |

**Risk Management**

| **Risk** | **Probability** | **Impact** | **Mitigation Strategy** |
| --- | --- | --- | --- |
| CSS not responsive on all devices | Medium | Medium | Use media queries, test on mobile and desktop |
| JavaScript bugs in cart functionality | High | High | Use console.log() and browser dev tools |
| File path or hosting errors | Medium | Low | Use proper folder structure, test GitHub Pages |
| Time overrun due to learning curve | Medium | Medium | Use small milestones to track progress weekly |

**Communication Plan**

If you're working alone:

* Keep a development journal (optional)
* Use Git commit messages as a changelog

If you’re working with others or submitting to a teacher:

* Update progress weekly
* Ask for peer reviews or instructor feedback after each milestone

**Summary**

The Planning Phase is one of the most critical stages of the Software Development Life Cycle (SDLC), even for a small-scale project like a basic food ordering website. It provides a structured approach to organizing the work, estimating timelines, choosing technologies, and identifying potential challenges in advance.

This phase ensures that the project has a **clear direction** and avoids last-minute confusion or mismanagement. Even in the absence of complex backend systems or large development teams, planning helps maintain focus and discipline throughout the development process.

For this specific project — a **simple front-end food ordering website** — the planning phase included:

* **Defining the project scope** clearly by identifying what is included (HTML structure, CSS design, JavaScript cart functionality) and what is excluded (e.g., backend server, payment gateway).
* **Estimating the timeline** realistically and breaking it into weekly milestones that made the tasks manageable and measurable.
* **Identifying tools and technologies** that are appropriate for a beginner to intermediate-level project, while ensuring the chosen stack is lightweight and effective.
* **Recognizing risks early** and planning basic strategies to address them (like using browser dev tools for debugging).
* **Simplifying team structure**, assuming a solo development approach where a single developer handles all responsibilities — from planning and development to testing and deployment.

This careful and thoughtful planning phase serves as a **foundation for successful project execution**, ensuring that all efforts align with the intended goals. It saves time, improves the quality of the final output, and makes the development process more enjoyable and less stressful.