# **Foodies-Hub: SDLC for a Food Blogging Page**

## **1. Planning (Understanding the Goal)**

### **Purpose of the Project:**

* **Foodies-Hub** will be a platform where food enthusiasts can share recipes, restaurant reviews, food photography, and cooking tips.
* **Goal**: To create a modern, user-friendly food blog with rich content and interactive features (such as recipe cards, comment sections, and photo galleries).

### **Problem the Project Solves:**

* Food lovers struggle to find a single platform that offers well-organized recipes, reviews, and food discussions in an easy-to-navigate format.

### **Team Roles:**

* **Jyothsna- Project Head**
* **Nandita- Designing**
* **Monika- Designing**
* **Mahima- Jss Developer**
* **Vidhita Gupta- Content Writer**

### **Major Milestones:**

* **Week 1**: Home Page Design & Basic Blog Structure
* **Week 2**: Recipe Page & Comment System
* **Week 3**: Photo Gallery and Review Integration
* **Week 4**: Deployment and Testing

## **2. Requirement Analysis (Defining What is Needed)**

### **Functional Requirements:**

* **Home Page**:  
  + A clean, appealing landing page with featured blog posts.
  + Navigation menu with links to different sections (Recipes, Reviews, Gallery, etc.).
  + Search bar for easy access to recipes and posts.
* **Recipe Page**:  
  + Individual posts for each recipe with ingredients, steps, and a photo gallery.
  + Option for users to leave comments or reviews.
* **Review Section**:  
  + Users can submit restaurant reviews with ratings and a comment section.
* **Comment System**:  
  + Visitors can comment on recipes and blog posts (use Disqus or a custom solution).

### **Non-Functional Requirements:**

* **Responsive Design**: The website should work on all screen sizes (desktop, tablet, mobile).
* **SEO Optimization**: Ensure pages are optimized for search engines for higher visibility.
* **Performance**: Optimize loading times for faster user experience (use image compression, minified CSS/JS).

### **Technical Constraints:**

* Hosting on **Netlify** or **Heroku** (for simplicity).
* Using a **CMS** (Content Management System) like WordPress or custom build using **Node.js/Express**.

## **3. Design (Creating a Blueprint of the Project)**

### **Wireframe:**

Create a wireframe for each major section:

* **Home Page**: Featured posts section, top navigation, footer with contact details and social media links.
* **Recipe Page**: Ingredients list, instructions, cooking tips, a comment section.
* **Gallery Page**: Grid layout for food photography with hover effects.

You can use tools like **Figma** or **Balsamiq** to design the wireframe, or even sketch the layout on paper.

### **UI Design:**

* **Color Scheme**: Use warm, appetizing colors like yellows, oranges, and browns for a food-related aesthetic.
* **Typography**: Use clean, readable fonts like **Roboto** or **Open Sans** for text-heavy sections; stylish fonts for headers like **Playfair Display**.
* **Images**: High-quality images of food for blogs and recipes.

### **Folder Structure:**

* index.html – Home Page
* recipe.html – Individual recipe pages
* reviews.html – Restaurant review section
* gallery.html – Photo gallery page
* style.css – Styling for the website
* script.js – JavaScript for interactive elements
* images/ – Folder for all images and media files
* assets/ – Any additional resources, like fonts or external libraries

## **4. Implementation (Coding the Project)**

### **Front-End Development:**

* **HTML**: Structure the pages with proper semantic tags (header, section, footer).
* **CSS**: Use Flexbox or CSS Grid for layout, along with media queries for responsiveness.
* **JavaScript**: Implement interactive features, like a search bar, image sliders for the gallery, and comment section functionality.

### **Version Control:**

* Use **Git** to create a repository on **GitHub**.
* Commit frequently with meaningful messages (e.g., "Initial commit with basic layout").

## **5. Testing (Ensuring the Project Works Correctly)**

### **Testing Types:**

* **Unit Testing**: Test individual components (e.g., recipe submission form, comment system).
* **Integration Testing**: Ensure the comment system works with each recipe page.
* **User Testing**: Conduct tests with real users to gather feedback on the site’s usability.

### **Test on Multiple Devices:**

* Ensure the website is fully responsive on desktop, mobile, and tablet.
* Test on major browsers (Chrome, Firefox, Safari).

### **Bug Tracking:**

* Log issues using **GitHub Issues** (e.g., "Fix button alignment in mobile view").
* Regularly address issues and ensure fixes are thoroughly tested.

## **6. Deployment (Making the Project Live)**

### **Hosting:**

* Use **Netlify** for deploying the static files (HTML, CSS, JS).
* For back-end (if using a custom CMS or server), deploy on **Heroku**.

### **Deployment Steps:**

* Push all changes to the **GitHub repository**.
* Set up deployment through **Netlify** or **Heroku**.
* Test the live site for broken links, missing images, and other issues.

### **Domain:**

* Buy a domain name (e.g., **foodies-hub.com**) and link it to your hosting platform.

## **7. Maintenance & Updates (Improving the Project Over Time)**

### **Regular Updates:**

* Add new recipes, blog posts, and reviews each month.
* Regularly update the photo gallery with high-quality food photography.

### **Performance Optimizations:**

* Compress images for faster loading times.
* Minify CSS and JavaScript for reduced file sizes.

### **User Feedback:**

* Collect feedback through surveys or a simple form.
* Incorporate suggestions for features and improvements.

## **GitHub Action Items for Students**

1. **Create a GitHub Repository**:  
   * Name it **foodies-hub** and provide a short project description.
2. **Set Up a README**:  
   * Write a description of the website and instructions for setting it up locally.
3. **Document the Project**:  
   * Maintain a /docs folder to store design documents, wireframes, and additional notes.
4. **Use Issues for Task Tracking**:  
   * Create issues for each new feature (e.g., "Add recipe submission page") or bug fix.
5. **Work with Branches**:  
   * Create feature branches (e.g., feature-recipe-page) and merge to the main branch after peer reviews.
6. **Commit Code Regularly**:  
   * Write meaningful commit messages, like "Added recipe page layout".
7. **Push and Pull Changes**:  
   * Use git push to upload your local changes and git pull to sync the latest changes from the main branch.

**Conclusion**

With the Foodies-Hub website, the goal is to create a dynamic, engaging platform for food enthusiasts. By following the SDLC process and utilizing GitHub for version control and collaboration, students can ensure the development process is well-organized and efficient.