SUP'COM

Higher School of Communication of Tunis

PROJET WEB

OBJECTIF-02

DÉVELOPPEMENT ET INTÉGRATION DES FONCTIONNALITÉS PRINCIPALES

Dr. Maroua Nouioua

Année universitaire 2025/2026

CONTEXTE

- Après avoir défini la structure (sitemap), conçu les prototypes et validé le design (Objectif 2), vous passez maintenant à la phase de développement concret.
- Cette étape vise à traduire vos maquettes en un site fonctionnel à l'aide du framework choisi et à implémenter les fonctionnalités essentielles qui répondent aux besoins définis lors de l'objectif 1.

MISE EN PLACE DE L'ENVIRONNEMENT DE DÉVELOPPEMENT

- Avant de coder, assurez-vous que votre environnement est prêt :
- Installez les outils nécessaires : IDE (VS Code), Node.js, npm/yarn, Git, navigateur,…
- Initialisez votre projet avec le framework choisi :

```
npx create-react-app myproject     # React
ng new myproject                   # Angular
vue create myproject               # Vue.js
```

- Mettez en place un repository GitHub/GitLab partagé avec vos coéquipiers.

→adoptez dès le début une bonne organisation des dossiers (src/, components/, assets/, services/, etc.) et des conventions de nommage claires.

INTÉGRATION DU DESIGN (FRONTEND)

- Transformez vos prototypes (réalisés à l'Objectif 2) en pages réelles :
  - Reproduisez la structure HTML selon le sitemap.
  - Appliquez vos styles CSS/SCSS ou utilisez un framework (Tailwind, Bootstrap, Material UI…).
  - Implémentez une navigation dynamique (menu, routes, liens entre pages).
  - Assurez la responsivité (mobile, tablette, desktop).
- Objectif clé : obtenir un site visuellement fidèle à votre maquette et fluide dans l'expérience utilisateur.

DÉVELOPPEMENT DES FONCTIONNALITÉS PRINCIPALES

- Ajoutez la logique fonctionnelle nécessaire au cœur de votre site :

➢Formulaires dynamiques : contact, inscription, recherche, etc.

➢Interactions utilisateur : boutons, animations, notifications, filtres.

➢Gestion de données (mock ou vraie API selon votre choix) :

➢Consommer une API REST existante (ex. JSONPlaceholder).

➢Ou créer un backend minimal (Node.js, Django, Laravel…) pour gérer vos données.

➢Gestion des états (React Hooks, Redux, ou équivalents selon framework).

→implémentez d'abord les fonctionnalités prioritaires, puis ajoutez des extras seulement si le cœur du site fonctionne parfaitement.

COLLABORATION ET BONNES PRATIQUES

- Travaillez en équipe agile : répartissez les tâches (frontend, backend, design, intégration).
- Utilisez Git pour le suivi des versions (commit clair, branche par fonctionnalité).
- Testez chaque composant avant l'intégration finale.
- Documentez vos choix techniques et vos dépendances.

→Simulez un vrai projet professionnel

→Communiquez via un outil comme Trello, Notion, ou GitHub Projects pour suivre vos tâches.

PRÉVISUALISATION ET TESTS

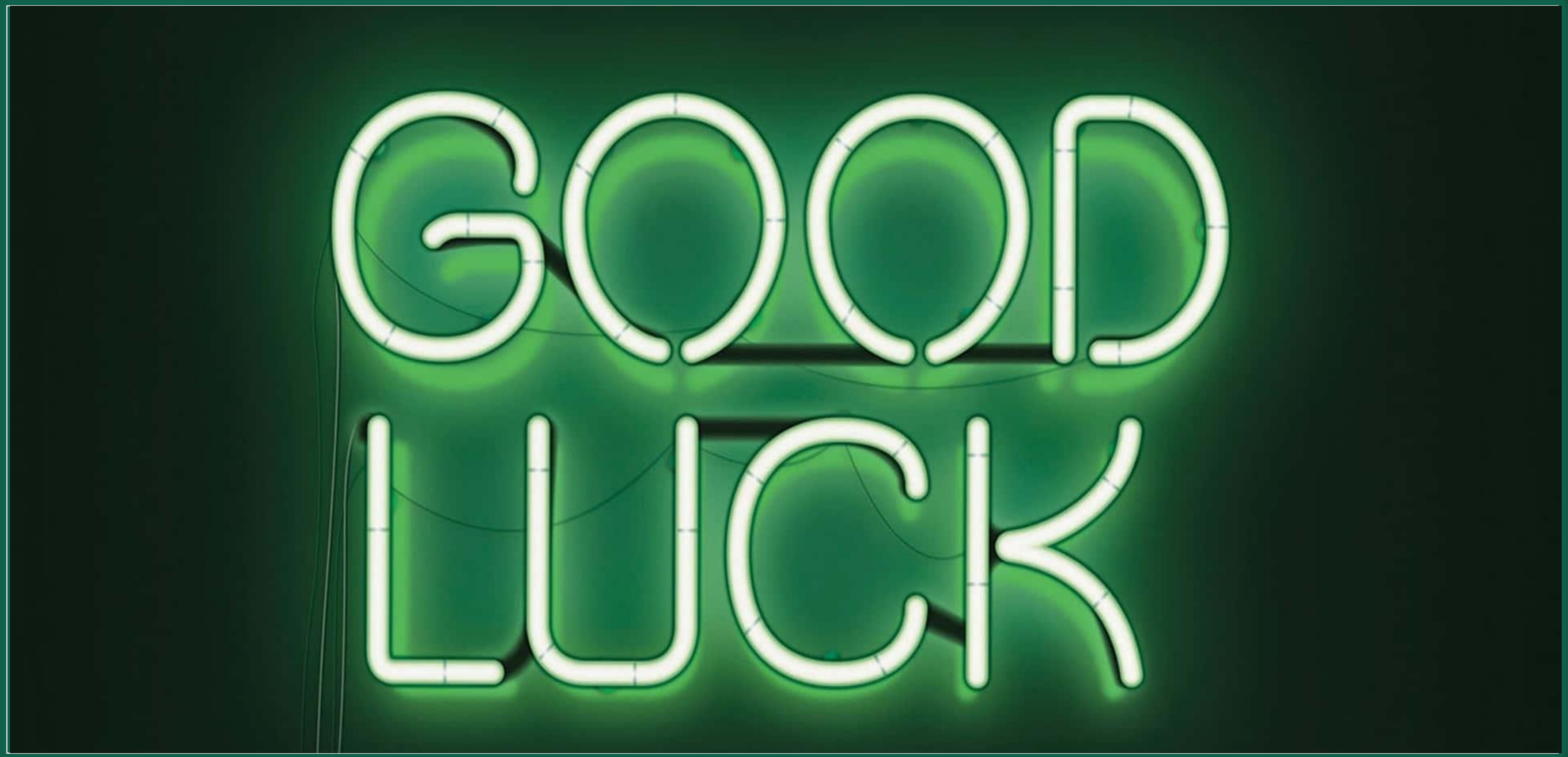
- Vérifiez la compatibilité multi-navigateurs (Chrome, Firefox, Edge, Safari).
- Testez les liens, formulaires et interactions.
- Optimisez les performances (poids des images, chargement rapide).
- Vérifiez la structure du code et l'accessibilité (balises sémantiques, contrastes).

LIVRABLE ATTENDU

- Format : dossier compressé .zip ou lien GitHub
- Contenu minimum :
  - Site web fonctionnel avec au moins 3 à 5 pages dynamiques
  - Frontend complet basé sur le framework choisi
  - Backend minimal ou simulation d'API
  - Documentation succincte (README) expliquant :
  - Le choix du framework
  - Les fonctionnalités développées
  - Les étapes de lancement du projet
- Nom du fichier : GroupeX_Projet_Web_Objectif3.zip ou lien GitHub

CONSEILS

- Testez régulièrement votre site dès les premières lignes de code.
- Validez vos choix techniques en équipe avant de les figer.
- Ne cherchez pas à tout faire : priorisez les fonctionnalités qui apportent une vraie valeur.
- N’oubliez pas que la qualité du code (lisibilité, structure, propreté) est aussi importante que le rendu visuel.


GOOD
LUCK