## ****2. Pages & Their Features****

### ****Page 1: Landing / Home Page****

* **Title:** “GreenChain – Empowering Africa’s Carbon Economy”
* **Features:**
  + **Hero Section:** A bold headline, subheading explaining the mission (e.g., “Blockchain-powered carbon credits for farmers & businesses”).
  + **Call-to-Action (CTA):** A prominent “Connect Wallet” button.
  + **Brief Overview:** A short paragraph or infographic about what GreenChain does.
  + **Navigation Bar:** Links to About, Marketplace, Dashboard, FAQ.
  + **Footer:** Contact info, social media links, and basic legal information.

### ****Page 2: About Page****

* **Title:** “About GreenChain”
* **Features:**
  + **Mission & Vision:** Clear statements about what the startup aims to achieve (e.g., empowering small-scale farmers).
  + **Problem & Solution:** A brief section describing the challenges in the current carbon market and how GreenChain addresses them.
  + **Timeline/ Roadmap:** Key milestones (even if in testing phase now, you can outline future goals).
  + **Team Section:** Brief bios and roles of founders (e.g., “Elly Logan – Frontend & Blockchain,” “Ian Kahuria – Backend & Infrastructure”).

### ****Page 3: Marketplace****

* **Title:** “Marketplace”
* **Features:**
  + **Listings:** A grid or list of carbon credit offerings. Each listing can be a card showing project details, available credits, price, etc.
  + **Search & Filter:** Options to sort by project type, price, verification status.
  + **Action Buttons:** If a seller is logged in, an option to list new carbon credits (which might interact with the smart contract).
  + **Info Panel:** Brief information on how the carbon credits are verified and the benefits of transparency.

### ****Page 4: Dashboard / User Portal****

* **Title:** “Dashboard”
* **Features:**
  + **Wallet Integration:** Show connected wallet address and current token balance. Use Web3.js/Ethers.js to fetch data.
  + **Transaction History:** Display recent transfers, staking activities, and rewards claimed.
  + **Staking Interface:**
    - **Stake Tokens:** A form where users input the amount to stake.
    - **Unstake Tokens:** A form/button to withdraw staked tokens.
    - **Claim Rewards:** Button to claim accumulated rewards.
  + **Interactive Cards/Graphs:** Visual display of staked tokens over time and reward accumulation (optional but engaging).

### ****Page 5: Admin Panel (Optional)****

* **Title:** “Admin Panel”
* **Features:**
  + **Restricted Access:** Only accessible if the connected wallet is the admin wallet.
  + **Admin Actions:**
    - Update community fund address.
    - Adjust community fee percentage.
    - Mint new tokens or withdraw from the community fund.
  + **Data Panels:** Show overall system metrics (e.g., total tokens staked, overall community fund balance).

### ****Page 6: FAQ / Help****

* **Title:** “Help & FAQ”
* **Features:**
  + **Frequently Asked Questions:** Common questions about how to connect a wallet, how staking works, etc.
  + **Tutorials/Guides:** Step-by-step instructions (text or video) on how to use the platform.
  + **Contact Form:** Option to reach out for support.

## ****3. Additional Components & Considerations****

**Header & Navigation Bar:**

* + Present on all pages. Includes links to Home, About, Marketplace, Dashboard, FAQ, and optionally Admin.
  + A “Connect Wallet” button that triggers MetaMask integration.

**Footer:**

* + Include basic links (Privacy Policy, Terms, Contact Info, Social Media).

**Reusable UI Components:**

* + **Cards:** For displaying marketplace listings and user activity.
  + **Forms:** For staking, transfers, and admin updates.
  + **Buttons:** Standardized buttons for actions.
  + **Modal Dialogs:** For confirmation prompts (e.g., “Are you sure you want to stake X tokens?”).

**Responsive Design:**

* + Ensure the UI works on both desktop and mobile devices.
  + Use CSS frameworks or custom media queries.

**Testing & Integration:**

* + Integrate smart contract calls using Ethers.js/ Web3.js to interact with your deployed contract.
  + Use React hooks (e.g., useState, useEffect) to manage state and update the UI on blockchain events.

## ****4. Suggested Navigation Flow****

1. **Landing Page ("/")**
   * Users land here, see an overview, and are prompted to connect their wallet.
2. **About ("/about")**
   * Provides context about the project.
3. **Marketplace ("/marketplace")**
   * Buyers and sellers can view and list carbon credits.
4. **Dashboard ("/dashboard")**
   * After connecting their wallet, users see their personal token data, staking interface, and transaction history.
5. **FAQ ("/faq")**
   * Additional help and support.
6. **(Optional) Admin Panel ("/admin")**
   * Protected route that checks if the connected wallet is ad