**User Stories & Acceptance Criteria**

1. **Climate connect system**
   * **User Story:** As a climate researcher, I want to analyze historical and real-time climate data so that I can predict future trends and assess risks.
   * **Acceptance Criteria:**
     1. System fetches climate data from APIs (NOAA, NASA, OpenWeatherMap).
     2. Machine learning model processes data and generates predictions.
     3. Dashboard visualizes climate trends and predictions.
2. **Smart Air Quality Monitoring System**
   * **User Story:** As a city planner, I want to monitor air quality in different locations so that I can implement pollution control measures.
   * **Acceptance Criteria:**
     1. IoT sensors collect real-time air quality data.
     2. AI model predicts air pollution trends.
     3. Dashboard displays air quality levels and sends alerts for hazardous conditions.

**Use Case Development**

1. **Title:** Climate connect
   * **Primary Actor:** Climate Researcher
   * **Secondary Actors:** Government agencies, Environmental NGOs
   * **Preconditions:** System has access to climate APIs and historical data.
   * **Basic Flow:**
     1. User selects climate parameters (temperature, humidity, CO2 levels, etc.).
     2. System fetches data from APIs.
     3. Machine learning model processes data.
     4. System generates and displays predictions.
     5. User reviews and downloads the report.
   * **Alternate Flow:** API is unavailable, and system notifies the user.
2. **Title:** Monitor Air Quality
   * **Primary Actor:** City Planner
   * **Secondary Actors:** Residents, Health Agencies
   * **Preconditions:** IoT sensors are active and transmitting data.
   * **Basic Flow:**
     1. IoT sensors collect air quality data.
     2. AI processes and predicts pollution levels.
     3. Dashboard displays current and predicted air quality.
   * **Alternate Flow:** Sensor failure, system notifies maintenance team.

**Data Modeling**

**Entities:**

* **Climate Data (ID, Date, Temperature, Humidity, CO2 Levels)**
* **Air Quality (ID, Location, Date, AQI, Pollutants)**
* **User (ID, Name, Role, Contact Info)**
* **Notifications (ID, UserID, Type, Message, Timestamp)**

**CRUD Matrix:**

| **Entity** | **Create** | **Read** | **Update** | **Delete** |
| --- | --- | --- | --- | --- |
| Climate Data | Yes | Yes | No | No |
| Deforestation Alert | Yes | Yes | Yes | No |
| Air Quality | Yes | Yes | Yes | No |
| User | Yes | Yes | Yes | No |
| Notifications | Yes | Yes | No | Yes |

**Nonfunctional Considerations**

1. **Performance:** The system must handle large climate datasets and process them efficiently to provide real-time insights.
2. **Scalability:** The system should be able to scale to accommodate additional climate data sources and new regions.
3. **Security:** Data encryption must be implemented to protect sensitive environmental data.
4. **Reliability:** The system should ensure a 99.9% uptime to provide continuous monitoring and analysis.
5. **Usability:** The interface should be user-friendly, allowing users to easily interpret environmental insights and predictions.

**Business Requirements:**

1. **Educate users and industries on climate change** through AI-driven learning modules.
2. **Enable interactive engagement**:
   * Twitter-like discussion forum for insights and solutions.
   * Corporate pledge & ranking system to track companies’ climate commitments.
3. **Provide real-time climate data** via APIs on:
   * Carbon emissions
   * Deforestation rates
   * Extreme weather patterns
4. **Monitor industry emissions** with:
   * Real-time emissions dashboard for various sectors.
   * Corporate sustainability index.
   * AI-powered predictive analytics for environmental impact forecasts and green policy recommendations.
5. **AI Chatbot for Climate Assistance**.

**Functional Requirements:**

1. **AI-driven learning modules** for climate education.
2. **Real-time data APIs**:
   * Carbon emissions data
   * Deforestation rates data
   * Extreme weather patterns data
3. **Emissions monitoring dashboard** for tracking sector-specific emissions.
4. **Corporate sustainability index** for ranking based on eco-friendly efforts.
5. **Predictive analytics** for environmental impact forecasts and policy recommendations.
6. **Discussion forum** for user and expert interactions.
7. **Corporate pledge tracking system** for climate commitments.
8. **AI Chatbot** for real-time climate assistance.

**External Requirements:**

1. **Data Sources**: Integration with external climate data providers for real-time updates.
2. **Industry Standards**: Compliance with environmental regulations and standards.
3. **Third-Party Services**: Use of third-party services for data analytics, API management, etc.

**Constraints:**

1. **Budget**: Limitations on project funding and resources.
2. **Timeline**: Project deadlines and delivery schedules.
3. **Data Privacy**: Adherence to data protection and privacy laws.
4. **Technological Limitations**: Constraints posed by current technology and infrastructure.

**Validation Checklist**

**Functional Requirements:**

1. Does the system fetch climate data from external APIs (NOAA, NASA, OpenWeatherMap)?
2. Does the machine learning model process climate data and generate predictions accurately?
3. Does the dashboard visualize climate trends and predictions in an easy-to-understand format?
4. Does the IoT sensor system collect real-time air quality data?
5. Does the AI model predict air pollution trends with a reasonable accuracy margin?
6. Does the system provide real-time air quality alerts for hazardous conditions?
7. Does the emissions monitoring dashboard track sector-specific emissions accurately?
8. Does the corporate sustainability index rank companies based on their eco-friendly efforts?
9. Does the AI chatbot provide real-time climate assistance and relevant insights?
10. Does the discussion forum facilitate engagement and knowledge sharing on climate issues?

**Non-functional Requirements:**

1. Does the system process large climate datasets efficiently?
2. Does the system scale to accommodate additional climate data sources and new regions?
3. Are data encryption and security protocols in place to protect sensitive environmental data?
4. Does the system maintain 99.9% uptime for continuous monitoring and analysis?
5. Is the user interface designed for accessibility and ease of use?

**Peer Review Plan**

**Roles:**

* **Moderator:** Ensures the review process stays on track.
* **Author:** Presents the developed feature.
* **Reviewers:** Evaluate the system for completeness and correctness.

**Checklist:**

1. Are all edge cases handled (e.g., API unavailability, sensor failure)?
2. Is there a fallback mechanism for missing or delayed data?
3. Does the system comply with relevant environmental regulations and data privacy laws?

**Hypothetical Issue Found:**

* Missing requirement to notify researchers when API data is outdated or unavailable.

**Prototyping Strategy**

* **Mockup (Horizontal Prototype)** is more suitable for validating the user interface (e.g., dashboard layout, climate visualization, air quality alerts).
* **Proof of Concept (Vertical Prototype)** is necessary for backend integration challenges, such as API data retrieval, machine learning model performance, and IoT sensor data processing.

**Traceability Matrix**

| **Requirement ID** | **Description** | **Source** | **Implementation** |
| --- | --- | --- | --- |
| Feature-01 | Fetch climate data from APIs | Climate Researchers | API Integration Module |
| Feature-02 | Process climate data using ML | Climate Researchers | ML Model |
| Feature-03 | Display predictions on dashboard | Climate Researchers | UI, Visualization Module |
| Feature-04 | Monitor air quality in real time | City Planners | IoT Sensor Network |
| Feature-05 | Predict pollution trends | City Planners | AI Model |
| Feature-06 | Provide real-time air quality alerts | Residents, Health Agencies | Alert Notification System |
| Feature-07 | Track corporate sustainability | Environmental NGOs | Sustainability Index |

**Tool Selection**

1. **Figma (Prototyping Tool)**
   * Purpose: UI/UX design and validation.
   * Usage: Create interactive mockups for the dashboard, chatbot, and discussion forum to validate user flows.
2. **Postman (API Testing Tool)**
   * Purpose: Validate API integration and data retrieval.
   * Usage: Test API calls to ensure data is fetched, processed, and displayed correctly.

**Feasibility Assessment**

* **Technical Feasibility:**
  + Integrating IoT sensors with AI predictions is complex but feasible with proper data processing pipelines.
  + Machine learning models require large datasets, making cloud computing resources necessary.
* **Resource Feasibility:**
  + Requires a skilled team in AI, IoT, and full-stack development.
  + Timeline must accommodate iterative testing and API reliability checks.

**Risk & Mitigation Strategy:**

* **Risk:** API data might be unavailable due to downtime or restrictions.
* **Mitigation:** Implement a caching mechanism to store the latest available data and notify users of outdated information.

This validation document ensures a structured approach to evaluating the system before deployment. Let me know if you need any refinements! 🚀