Here’s a breakdown of coding-related tasks for adding the greenhouse gas module

Here’s a breakdown of coding-related tasks for adding the greenhouse gas module to the People in Ecosystem and Watershed Integration (PEWI). The tasks are structured into *phases* and *roles*, ensuring each builds on the previous ones. I know some phases might overlap, but that is okay. The most important thing is getting a working document to guide the development process. please le tme know all your thoughts

Phase 1: Initial Setup and Planning

Team Roles:

- Define roles (e.g., back-end, front-end, data management).
- Assign tasks to individual team members.

Tasks:

1. **Set Up Project Environment:**
    - Create a dedicated repository/new branch for the greenhouse gas (GHG) module within PEWI.
    - Ensure adherence to coding conventions and version control practices.
    - Define module architecture and interfaces with existing system components.
    - Brainstorm module requirements, including GHG accounting, UI, and data integration.
    - We discussed adding a data ribbon and the GHG summary to the existing data summary data in PEWI. TThese and a lot more need to be discussed in details
2. **Define Data Structures:**
    - Choose data structures for tracking methane (CH_4) and nitrous oxide (N_2O) emissions (CSV or SQL database, preference for SQL).
    - Clearly define interfaces for data exchange between the GHG module and other PEWI components.

Outcome:

- A well-defined project structure with clear roles, responsibilities, and an architectural plan for the module.

Phase 2: Development of Core Functionality

Team Roles:

- Back-end developers focus on data integration and GHG simulation logic.

- Data specialists handle sourcing and processing data for simulations.
- Front-end developers design UI components.

Tasks:

1. **Develop GHG Accounting Framework:**
   - Create a prototype for the GHG accounting framework.
   - Implement input/output functions for GHG simulation.
2. **Develop Simulation Logic:**
   - Link GHG calculations to land use and other environmental data (e.g., soil type, fertilizer data).
   - Write unit tests to validate GHG simulation logic.
3. **Data Integration:**
   - Write functions to pull and process relevant data for methane and nitrous oxide emissions.
   - Ensure the GHG data is linked to each land use type.

Outcome:

- Functional GHG accounting framework and core simulation logic in place, with integrated data.

Phase 3: User Interface and Visualization Development

Team Roles:

- Front-end developers focus on user experience and visualization.
- Back-end developers ensure compatibility between UI and GHG calculations.
  Tasks:

1. **Design User Interface Components:**
   - Create user-friendly interface elements for inputting GHG simulation parameters.
   - Integrate interactive visualizations for methane and nitrous oxide emissions, such as charts and graphs.
   - We have to decide how we need the data to be visualized and retrieved by the user
2. **Develop Visualization Tools:**
   - Implement functions to generate clear visualizations of methane and nitrous oxide emissions over time.
   - Ensure compatibility with PEWI's existing visualization tools.

Outcome:

- A working user interface for inputting data and viewing simulation results, with interactive GHG visualization capabilities.

Phase 4: System Integration and Testing

Team Roles:

- Testing focus on system-wide testing, including compatibility with various browsers and operating systems
- All developers collaborate on refining and debugging.
- Developers are required to openly report all bugs they encounter, including those that were ignored or deemed non-critical, as well as any unresolved issues

Tasks:

1. **Integration with PEWI System:**
    - Ensure the GHG module is integrated with PEWI, including data flow between components. such as different existing modules and land uses
2. **System-Wide Testing:**
    - Conduct system-wide testing using complete datasets to ensure functionality and performance.
3. **Refactoring:**
    - Refactor code to improve efficiency, scalability, and maintainability.

Outcome:

- Fully integrated and tested GHG module within PEWI, ready for user deployment.
- Discuss and review the changes

Phase 5: Documentation and Finalization

Team Roles:

- Each developer is responsible for thoroughly documenting the components they have developed.
- The entire team is responsible for reviewing these documents to ensure clarity, logical flow, and completeness

Tasks:

1. **User Documentation:**
    - Develop clear and concise documentation for users to utilize the GHG module effectively.
    - Create readme-files in the respective project folders
    - blog the new development to the project website or LinkedIn page
2. **Final Testing and Review:**
    - Conduct final testing and full datasets.
    - invite third party to give feedback on the design and development status

- Ensure the module is ready for deployment.

Outcome:

- Comprehensive user documentation, final review, and fully functional GHG module.
- All data schemas declared in the project repository or data containers for future developers
- Push all changes to the master and send to the server
- Suggest future development changes and tasks
- Share experience and tactics

Summary of Tasks:

1. **Initial Setup and Planning (Phase 1)**: Define roles, set up the project environment, decide on architecture, and establish data structures.
2. **Core Development (Phase 2)**: Develop the GHG framework and simulation logic and integrate data.
3. **UI and Visualization (Phase 3)**: Create the user interface and develop visualization tools.
4. **System Integration and Testing (Phase 4)**: Integrate the module into the PEWI system and conduct system-wide testing.
5. **Documentation and Finalization (Phase 5)**: Document the system, finalize testing, and ensure readiness for deployment.