Protocol for Developing an Online Research Tool

Creating an online tool that is useful for individuals to study their living environment and for scientists to monitor societal phenomena is a multi-faceted project that involves several key steps. Following the underneath steps in the protocol ensures a systematic approach to developing an online tool for studying the living environment. Each step is crucial for creating a tool that is functional, user-friendly, and provides valuable insights to its users. Regular feedback and updates will keep the tool relevant and effective.

1. **Describe the questions, taking into account the dual purpose of the tool**
   - Establish partnerships with societal partners and data providers. Try to build these partnerships in a sustainable manner, so that you, your successors, and colleagues reap the benefits from the partnership for a longer period of time, rather than for the duration of the project.
   - Identify the societal challenge together with the societal partners.
   - Identify the research question that connects to the societal challenge.
   - Review scientific literature and existing tools.
   - Develop a project plan with timelines and milestones.

2. **Define Objectives and Requirements**
   - Identify Goals: determine what aspects the tool will study (e.g., presence at meetings, motivations about membership, energy use, housing quality, noise levels, water quality, green space).
     → Conduct surveys or interviews with potential users.
   - User Needs: understand the needs and expectations of the target users (e.g., ease of use, types of data, visualization preferences).
     → Define key performance indicators (KPIs) to measure the success of the tool.

3. **Research and Data Collection**
   - Data Sources: identify reliable sources of environmental data (e.g., government databases, scientific research, crowdsourced data).
   - Data Types: determine the types of data needed (e.g., real-time data, historical data, satellite imagery).

4. **Design and Planning**
   - User Interface (UI) Design: create a user-friendly interface that allows easy interaction with the tool. Keep in mind which environment you're building the tool for: desktop, mobile, both?
   - User Experience (UX) Design: ensure the tool is intuitive, engaging, and accessible.
   - Technical Architecture: plan the technical infrastructure, including server requirements, database design, and API integration. Keep in mind the costs for keeping the tool running. API's can adopt a single payment or subscription plan and server costs can be incredibly expensive. Since CollectieveKracht can't guarantee to cover these costs, it would be wise to factor future developments into the technical architecture. It is also worthwhile to make arrangements about who covers the costs to keep this tool running should there be a sudden change in the bills to be paid.
   - Create wireframes (visual guide that represents the skeletal framework of the tool) and mockups (CK).
   - Ensure compliance with data privacy laws and regulations (CK)
   - Keep in mind which design aspects you will leave out (based off of the objectives and requirements). Doing this also makes it easier to think about integrating future versions of the tool in the CK-platform.

5. **Development**
   - Front-End Development: build the user interface using appropriate web technologies (e.g., HTML, CSS, JavaScript).
   - Back-End Development: develop the server-side logic, database interactions, and data processing capabilities.
   - API Integration: integrate third-party APIs for data collection (e.g., weather APIs, pollution data APIs).
   - Choose appropriate front-end frameworks and back-end technologies (CK). These may depend on the platform or server where your tool will be running. Make sure that the frameworks and technologies are compatible with these. With possible future changes in mind, it may be wise to choose frameworks that are as common as possible (e.g., HTML, CSS, JavaScript). Implement secure authentication and user management (CK).

6. **Data Processing and Analysis**
   - Data Cleaning: implement methods to clean and validate the data to ensure accuracy.
   - Data Analysis: develop algorithms to analyze the data and extract meaningful insights.
   - Visualization: create visualizations (e.g., charts, maps) to help users understand the data.
   - Use data analysis tools and libraries.

7. **Testing**
   - Unit Testing: test individual components for functionality.
   - Integration Testing: ensure that all components work together seamlessly.
   - User Testing: Conduct beta testing with real users to gather feedback and identify any usability issues.
   - Perform cross-browser testing to ensure compatibility (CK).
   - Gather user feedback through surveys and usability tests (CK).
   - Perform cookie-testing to see if and how cookies influence the tool.

8. **Deployment (CK)**
   - Integrate in CK as a hosting platform and deploy the tool.
   - If possible: set up monitoring tools to track performance, uptime, and user interactions.
   - Maintenance: plan for regular updates, bug fixes, and improvements based on user feedback.
   - Use continuous integration and continuous deployment (CI/CD) pipelines.
   - Optimize the tool for performance and speed.
   - Implement disaster recovery plans.

9. **User Support and Documentation (CK)**
   - Help Resources: provide comprehensive documentation, FAQs, and tutorials to help users navigate the tool.
   - Customer Support: offer channels for user support (e.g., email, chat support).
   - Create video tutorials and user guides.
   - Set up a community forum for user interaction.
   - Continuously update documentation based on user feedback.