Creating a web application with Python that uses PostgreSQL as a database and collects data from users can be accomplished using a variety of frameworks and libraries. Given the requirement to make the application "interesting," we can add engaging features, interactive elements, and dynamic content. Below, I propose an idea for a web application including a brief on technologies, design considerations, and features that might make it engaging and practical.

**Web Application Idea: Custom Travel Planner**

**Concept:** A Custom Travel Planner app that enables users to create personalized travel plans. The application would collect user preferences like budget, destinations of interest, type of activities (e.g., cultural, adventure, relaxation), dietary restrictions, and more. Based on these inputs, the application will suggest customized travel itineraries, book accommodations, and make recommendations for dining and activities at the destination.

**Key Features:**

1. **User Profile Management:**
   * Users can create profiles and specify their interests, preferences, and past travel experiences.
2. **Interactive Travel Plan Builder:**
   * A dynamic form that captures user preferences.
   * An interactive map where users can explore possible destinations and see highlighted attractions, restaurants, etc.
3. **Customized Itinerary Suggestions:**
   * Algorithms that generate travel itineraries based on user inputs and real-time data (like weather, local events).
   * Options to manually adjust itineraries.
4. **Booking Integration:**
   * Partnerships with booking services (hotels, flights, car rentals) integrated directly for one-stop itinerary booking.
5. **Community Features:**
   * Users can share their itineraries and experiences.
   * Rating and reviewing components for destinations and accommodations.
6. **Real-time Notifications:**
   * Alerts about flight deals, weather conditions, and local events.

**Technologies:**

**Back-end:**

* **Python** with **Flask or Django** as the web framework:
  + Flask provides simplicity and flexibility, suitable for a lighter, more customizable app.
  + Django offers a more all-in-one solution with its own ORM and admin panel, ideal for managing complex data interactions.
* **PostgreSQL** :
  + To store user profiles, travel preferences, itineraries, and community-generated content.
  + Utilize PostgreSQL's robust features like geographic data extensions (PostGIS) if location-based services are heavily used.

**Front-end:**

* **HTML/CSS/JavaScript** with frameworks like **React or Vue.js** for a dynamic and responsive UI.
* **Bootstrap** or **TailwindCSS** for styling and layout.

**APIs and Libraries:**

* **Google Maps API** for map integrations.
* External APIs for travel bookings, weather updates, etc.
* **Celery** with **Redis** for handling asynchronous tasks and real-time notifications.

**Development Approach:**

1. **Prototyping:**
   * Design wireframes and mockups for user interfaces.
   * Establish core functionalities and database schema.
2. **Development:**
   * Set up the back-end environment (Python environment, PostgreSQL database).
   * Develop API endpoints and integrate external services.
   * Build front-end components and ensure mobile responsiveness.
3. **Testing and Iteration:**
   * Perform unit tests, integration tests, and usability testing.
   * Collect feedback from beta testers and iterate based on feedback.
4. **Deployment:**
   * Use services like Heroku, AWS, or DigitalOcean for deployment.
   * Consider Docker for containerization and easy scalability.
5. **Maintenance and Updates:**
   * Regular database backups.
   * Updates based on user feedback and new requirements.

**Conclusion:**

A Custom Travel Planner web application allows users to plan trips in a personalized and comprehensive manner. By integrating various APIs and deploying a user-friendly interface, the application can serve as a useful tool for both novice and experienced travelers, further enriched by community-driven insights and real-time data integrations.