Step-by-Step Instruction Plan to Build the Boomerang Cycle Website and Mobile Apps

Phase 1: Preparation and Planning

1. **Finalize Pseudocode (Categories 1–15)**
   - Ensure completeness, accuracy, and alignment with the project vision.
   - Confirm each category covers all core functions (registration, donation processing, cycles, payouts, compliance).
2. **Define Technology Stack**
   - **Frontend**: HTML, CSS, JavaScript (React.js or Angular), plus responsive frameworks (Bootstrap/Tailwind).
   - **Backend**: Node.js, Python (Django/Flask), or PHP.
   - **Database**: PostgreSQL, MySQL, or MongoDB.
   - **Payment Gateway**: Stripe, PayPal, cryptocurrency APIs.
   - **Hosting/Infrastructure**: AWS, Google Cloud, or Azure.
3. **Create a Project Roadmap**
   - Define phased versions:
     - *Version 1 (MVP)*: Registration, donations, and UI.
     - *Version 2*: Core donation cycles and payouts.
   - Establish milestones, dependencies, and resource allocation.

4. **Set Up Development Environment**
   - Install IDEs (VS Code), DB management tools, and version control (Git).
   - Create a GitHub repository for collaboration and CI/CD integration.

Phase 2: Frontend Development (MVP: User Interface)

5. **Wireframes & Mockups**
   - Create designs with Figma or Adobe XD for homepage, registration, dashboard, donation history, etc.
   - Prioritize usability, accessibility, and responsive design.
6. **Develop UI Components**
   - Implement static layouts (HTML/CSS/JS).
   - Add placeholders for dynamic content (e.g., donation history).
7. **Implement User Registration & Login**
   - Build secure signup/login forms with validation.
   - Support email, social logins, and secure password reset.
   - Connect to backend APIs for account creation and storage.
8. **Build Dashboard & Navigation**
   - Create a dashboard showing donation metrics, cycle status, and notifications.
   - Implement a sidebar/menu for easy navigation.
9. **Test Frontend**

- Verify responsiveness across devices.
- Ensure cross-browser compatibility.

Phase 3: Backend Development (Core Functionality)

10. **Backend Server Setup**
    - Configure Node.js/Django server, connect to DB, and define API endpoints.
11. **Donation Processing**
    - Translate pseudocode into code:
        - Accept $6.15 payments ($6 donation + $0.15 fee).
        - Split into six $1 micro-donations.
    - Queue donations for cycle assignment.
12. **Dynamic Interval Creation**
    - Implement logic for 1,440 intervals per 24h cycle.
    - Ensure 1,386 donors = 1 interval (6 × 231).
13. **Recipient Selection**
    - Algorithmically assign recipients using audited randomization.
    - Apply ratios: 1:231 standard, 1:50 premium.
14. **Multiple Cycles Handling**
    - Trigger new cycles when maximum interval/donor thresholds are reached.
    - Prevent overlaps or duplication.

15. **Integrate Payment Gateways**
    - Securely connect Stripe/PayPal/crypto APIs.
    - Validate and test full payment flows.

Phase 4: Testing and Debugging

16. **Unit Testing**
    - Verify individual modules (e.g., registration, donation split).
    - Use automated frameworks (Jest, PyTest).
17. **Integration Testing**
    - Confirm smooth interaction between frontend, backend, and DB.
18. **Load/Stress Testing**
    - Simulate high volumes (millions of donors).
    - Identify bottlenecks, latency, or scaling issues.
19. **Bug Fixing & Optimization**
    - Resolve defects and refine code for security, speed, and maintainability.

Phase 5: Deployment and Launch

20. **Deploy to Staging**
    - Launch staging environment for final QA and User Acceptance Testing (UAT).

21. **Prepare Production Infrastructure**
    - Configure servers, HTTPS, WAF/firewalls, and backups.
    - Establish monitoring and disaster recovery procedures.
22. **Official Launch**
    - Deploy production version.
    - Announce launch through social media, influencer campaigns, and email outreach.

Phase 6: Post-Launch Activities

23. **Monitor System Performance**
    - Track uptime, latency, donation throughput, and anomaly alerts.
24. **Collect User Feedback**
    - Use in-app surveys, reviews, and support tickets.
25. **Iterative Improvement**
    - Release regular updates addressing bugs and feature requests.
    - Continuously enhance transparency and user experience.

Key Deliverables by Phase

- **Phase 1**: Finalized pseudocode, tech stack, project roadmap, dev environment.
- **Phase 2**: UI with signup/login, dashboard, and navigation.

- **Phase 3**: Backend for donation cycles, processing, and payouts.
- **Phase 4**: Fully tested system.
- **Phase 5**: Production-ready website & app.
- **Phase 6**: Ongoing monitoring, scaling, and improvement.

Objective

Deliver a **global micro-donation platform** (web + mobile) that enables $6 donations to be fairly redistributed with full transparency, security, and compliance — not as a lottery, but as a **cycle of generosity**.

Functional Requirements

- **User Registration**: Email/social signups, secure login, 2FA, payout info collection.
- **Donation Processing**: $6 donations + $0.15 fee, six $1 micro-donations, secure payment gateways.
- **Cycle Management**: Dynamic intervals, fair allocation (1:231 standard, 1:50 premium).
- **Payout Distribution**: Transparent, audited, and trackable.
- **Premium Membership**: $100/year, enhanced ratio (1:50).
- **Localization**: Multi-language (29+) and multi-currency with real-time FX conversion.
- **Scalability**: Tens of millions of donors, multi-region architecture.
- **Security & Compliance**: Encryption, fraud detection, GDPR/AML compliance.
- **Transparency & Analytics**: Real-time dashboards, detailed reports.

- **Mobile Apps**: Native iOS/Android with push notifications.

Development Phases

- **Phase 1 (MVP)**: Registration, donations, cycles, simple dashboard.
- **Phase 2**: Premium membership, localization, advanced transparency features.
- **Phase 3**: Optimization, scaling, global compliance readiness.

Technical Stack

- **Frontend**: React.js / Angular
- **Backend**: Node.js / Python (Django/Flask)
- **Database**: PostgreSQL / MySQL / MongoDB
- **Mobile Apps**: React Native / Flutter
- **Hosting**: AWS / GCP / Azure
- **Payments**: Stripe, PayPal, Crypto APIs

Testing & Deployment

- **Testing**: Unit, integration, stress, and UAT.
- **Deployment**: Staging → Pilot Launch → Full Production.

Timeline & Milestones

- **Weeks 1–4**: Requirements & setup.
- **Weeks 5–12**: MVP development.
- **Weeks 13–16**: Testing & pilot.
- **Weeks 17–24**: Advanced features & scaling.

Next Steps

1. Confirm finalized pseudocode categories.
2. Define and lock the technology stack.
3. Begin Phase 1 (Preparation & Planning).
4. Kick off Phase 2 (Frontend MVP).

If you have any questions or improvements, please send an email to:
theboomerangcycle@gmail.com

Thank you, Eldad Rubin/