EcoTrack: Development Roadmap

🎯 **Project Overview**
Timeline: 12-14 weeks
Complexity: 8.5/10
Architecture: Laravel API + Vue SPA + Real-time features

📅 **Phase 1: Foundation & Setup**
Weeks 1-2 | Priority: Critical

Week 1: Environment & Core Setup

Day 1-2: Project Initialization
- [ ] Initialize Laravel 12 project
- [ ] Set up Vue 3 + Vite frontend
- [ ] Configure Docker environment (MySQL, Redis, Laravel, Nginx)
- [ ] Set up version control (Git) with proper .gitignore
- [ ] Configure environment files (.env setup)

Day 3-4: Database Foundation
- [ ] Create and run user migration with roles/status
- [ ] Set up Sanctum for API authentication
- [ ] Create User model with relationships
- [ ] Set up basic API routes structure
- [ ] Test basic authentication endpoints

Day 5-7: Frontend Foundation
- [ ] Configure Vue Router with basic routes
- [ ] Set up Pinia stores (auth, user)
- [ ] Create basic layout components (Header, Sidebar, Footer)
- [ ] Implement login/register forms
- [ ] Connect frontend auth to Laravel Sanctum
- [ ] Set up Axios interceptors and API utilities

🎯 Milestone 1: User can register, login, and access protected routes

Week 2: Core User Features

Day 8-10: User Management
- [ ] Complete user profile system
- [ ] Implement avatar upload with basic image processing
- [ ] Create user dashboard layout
- [ ] Build profile editing forms
- [ ] Add form validation (frontend + backend)

Day 11-14: Navigation & Settings
- [ ] Implement role-based navigation
- [ ] Create settings page (password change, preferences)
- [ ] Add language switching (EN/DE) infrastructure
- [ ] Implement basic error handling and loading states
- [ ] Set up Toast notifications system

🎯 Milestone 2: Complete user management system with file uploads

---

🌱 **Phase 2: Emissions Tracking System**
Weeks 3-4 | Priority: High

Week 3: Emissions Core

Day 15-17: Database & Models
- [ ] Create emissions table and migration
- [ ] Build Emission model with relationships
- [ ] Create EmissionCalculatorService
- [ ] Implement carbon footprint calculation logic
- [ ] Set up emission activity types and metadata

Day 18-21: Emissions API
- [ ] Build EmissionController with CRUD operations
- [ ] Create EmissionRequest validation
- [ ] Implement EmissionResource for API responses
- [ ] Add emission analytics endpoints
- [ ] Create background job for eco-score calculation

🎯 Milestone 3: Backend can handle emission logging and calculations

Week 4: Emissions Frontend

Day 22-24: Emission Forms
- [ ] Create EmissionForm component with dynamic fields
- [ ] Implement activity type switching logic
- [ ] Add real-time carbon calculation display
- [ ] Build emission list/history component
- [ ] Add emission editing and deletion

Day 25-28: Analytics & Insights
- [ ] Create emission analytics dashboard
- [ ] Implement Chart.js for data visualization
- [ ] Build emission trends and insights components
- [ ] Add date range filtering
- [ ] Create eco-score display and progress tracking

🎯 Milestone 4: Complete emissions tracking with analytics

👥 **Phase 3: Social Features Foundation**
Weeks 5-6 | Priority: High

Week 5: Posts & Content System

Day 29-31: Posts Database
- [ ] Create posts, reactions, follows tables
- [ ] Build Post, Reaction, Follow models
- [ ] Implement polymorphic reactions system
- [ ] Set up post status and content moderation
- [ ] Create tags system with JSON storage

Day 32-35: Posts API
- [ ] Build PostController with full CRUD
- [ ] Implement reaction system endpoints
- [ ] Create post feeds (latest, popular, following)
- [ ] Add image upload for posts

- [ ] Implement post reporting functionality

🎯 Milestone 5: Complete posts system with reactions

Week 6: Social Interactions

Day 36-38: Follow System
- [ ] Create SocialController for follow/unfollow
- [ ] Implement follower/following lists
- [ ] Build user profile pages
- [ ] Add user search and discovery
- [ ] Create follow suggestions logic

Day 39-42: Posts Frontend
- [ ] Build PostCard component with reactions
- [ ] Create PostForm with image upload
- [ ] Implement PostFeed with filtering
- [ ] Add user profile components
- [ ] Build follow/unfollow functionality

🎯 Milestone 6: Complete social interaction system

🔔 **Phase 4: Real-time & Notifications**
Weeks 7-8 | Priority: Medium-High

Week 7: Notification System

Day 43-45: Notifications Backend
- [ ] Create notifications table and model
- [ ] Build NotificationService
- [ ] Implement notification types and templates
- [ ] Create notification event listeners
- [ ] Set up email notification jobs

Day 46-49: Laravel Reverb Setup
- [ ] Install and configure Laravel Reverb
- [ ] Create broadcasting events (UserFollowed, PostReacted, etc.)
- [ ] Set up WebSocket authentication
- [ ] Test real-time broadcasting
- [ ] Implement notification broadcasting

🎯 Milestone 7: Real-time notification system working

Week 8: Frontend Real-time

Day 50-52: WebSocket Integration
- [ ] Set up Laravel Echo in Vue app
- [ ] Create useWebSocket composable
- [ ] Implement notification store with real-time updates
- [ ] Build notification dropdown component
- [ ] Add WebSocket connection management

Day 53-56: Real-time Features
- [ ] Implement real-time post feed updates
- [ ] Add live notification badges
- [ ] Create notification center page

- [ ] Add sound/visual notification alerts
- [ ] Test real-time features across different browsers

🎯 Milestone 8: Complete real-time notification system

🛡️ **Phase 5: Moderation & Admin**
Weeks 9-10 | Priority: Medium

Week 9: Content Moderation

Day 57-59: Reporting System
- [ ] Create reports table and model
- [ ] Build reporting functionality for users/posts
- [ ] Implement ModerationService
- [ ] Create moderator dashboard backend
- [ ] Add content moderation workflows

Day 60-63: Admin Backend
- [ ] Build admin user management endpoints
- [ ] Create announcement system
- [ ] Implement system analytics service
- [ ] Add bulk operations for users/content
- [ ] Create admin permission middleware

🎯 Milestone 9: Complete moderation and admin backend

Week 10: Admin Frontend

Day 64-66: Moderator Panel
- [ ] Create moderator dashboard layout
- [ ] Build content moderation interface
- [ ] Implement report management system
- [ ] Add user management tools
- [ ] Create moderation action logging

Day 67-70: Admin Panel
- [ ] Build admin dashboard with analytics
- [ ] Create user management interface
- [ ] Implement announcement management
- [ ] Add system monitoring views
- [ ] Create admin settings panel

🎯 Milestone 10: Complete admin and moderation interfaces

📊 **Phase 6: Advanced Analytics & Performance**
Weeks 11-12 | Priority: Medium

Week 11: Advanced Features

Day 71-73: Enhanced Analytics
- [ ] Build advanced emission analytics
- [ ] Create community comparison features
- [ ] Implement achievement system
- [ ] Add data export functionality

- [ ] Create personalized insights

Day 74-77: Performance Optimization
- [ ] Implement Redis caching strategies
- [ ] Add database query optimization
- [ ] Set up background job processing
- [ ] Implement image processing optimization
- [ ] Add API rate limiting

🎯 Milestone 11: Advanced analytics and performance optimization

Week 12: Testing & Polish

Day 78-80: Testing Implementation
- [ ] Write comprehensive feature tests
- [ ] Create unit tests for services
- [ ] Add frontend component tests
- [ ] Implement integration tests
- [ ] Set up CI/CD pipeline testing

Day 81-84: Final Polish
- [ ] Complete multi-language implementation
- [ ] Add comprehensive error handling
- [ ] Implement loading states and skeletons
- [ ] Optimize frontend performance
- [ ] Create deployment documentation

🎯 Milestone 12: Production-ready application

🚀 **Phase 7: Deployment & Documentation**
Weeks 13-14 | Priority: High

Week 13: Deployment Preparation

Day 85-87: Production Setup
- [ ] Configure production Docker setup
- [ ] Set up production database with optimizations
- [ ] Configure Redis for production
- [ ] Set up SSL certificates
- [ ] Configure backup strategies

Day 88-91: Deployment & Testing
- [ ] Deploy to production environment
- [ ] Run performance testing
- [ ] Conduct security testing
- [ ] Test real-time features in production
- [ ] Monitor application performance

Week 14: Documentation & Portfolio

Day 92-94: Documentation
- [ ] Create comprehensive API documentation
- [ ] Write deployment and setup guides
- [ ] Document architecture decisions
- [ ] Create user guide and features overview
- [ ] Write technical blog posts

Day 95-98: Portfolio Presentation
- [ ] Create project showcase materials
- [ ] Record demo videos
- [ ] Prepare technical presentation
- [ ] Update portfolio website
- [ ] Create case study documentation

🎯 Final Milestone: Complete portfolio-ready application

📋 **Task Dependencies & Critical Path**

Critical Dependencies:
1. **Authentication → All Features** (Week 1-2)
2. **Database Schema → All Data Features** (Week 1-3)
3. **Posts System → Social Features** (Week 5 → Week 6)
4. **Notification Backend → Real-time Frontend** (Week 7 → Week 8)
5. **User Roles → Admin Panel** (Week 2 → Week 9-10)

Parallel Development Opportunities:
- Frontend components can be built while backend APIs are being developed
- Testing can be written alongside feature development
- Documentation can be updated continuously

🛠️ **Daily Development Guidelines**

Daily Routine:
1. **Start:** Review previous day's code, check for any issues
2. **Plan:** Break down daily tasks into 2-4 hour chunks
3. **Develop:** Follow TDD when possible, commit frequently
4. **Test:** Test features as you build them
5. **Document:** Update documentation and comments
6. **Review:** End-of-day code review and planning for next day

Weekly Reviews:
- **Monday:** Plan week's objectives
- **Wednesday:** Mid-week progress check
- **Friday:** Weekly milestone review and next week planning

🎯 **Success Metrics**

Technical Metrics:
- [ ] All major features implemented and working
- [ ] 90%+ test coverage on critical paths
- [ ] Page load times < 2 seconds
- [ ] API response times < 500ms
- [ ] Real-time features working reliably

Portfolio Metrics:
- [ ] Demonstrates full-stack capabilities
- [ ] Shows modern web development practices
- [ ] Includes real-time and advanced features
- [ ] Professional documentation and presentation

- [ ] Deployable and production-ready

🚨 **Risk Management**

High-Risk Areas:
1. **Real-time Features (Week 7-8):** WebSocket setup can be complex
   - *Mitigation:* Start with polling fallback, extensive testing
2. **Image Processing (Week 2, 6):** File handling can be tricky
   - *Mitigation:* Use established libraries, implement error handling
3. **Performance (Week 11):** Complex queries might be slow
   - *Mitigation:* Regular performance testing, optimize as you go

Contingency Plans:
- **Behind Schedule:** Prioritize core features over nice-to-haves
- **Technical Blocks:** Have alternative approaches ready
- **Scope Creep:** Stick to roadmap, note additional features for future

---

📈 **Progress Tracking**

Weekly Milestones:
- Week 2: ✅ Basic user management
- Week 4: ✅ Emissions tracking system
- Week 6: ✅ Social features foundation
- Week 8: ✅ Real-time notifications
- Week 10: ✅ Admin and moderation
- Week 12: ✅ Testing and optimization
- Week 14: ✅ Production deployment

Daily Tracking:
Use a simple progress tracker:
- 🔴 Not Started
- 🟡 In Progress
- 🟢 Completed
- 🔵 Testing
- ⚪ Blocked

This roadmap gives you a clear path from zero to a portfolio-worthy application in 12-14 weeks. Each phase builds on the previous one, and the milestones ensure steady progress toward your goal of landing a remote job or position in Germany!