**Biweekly Report**

**Project Title**: Diagnosis of Diabetic Retinopathy System  
**Time Period**: April 21 – May 11, 2025  
**Team Member**: Peijin Chen  
**Working Hours**: 15 hours

**Project Progress Overview**

During this period, front-end development focused on **streamlining the system structure** and **optimizing core workflows**, and we delivered an initial interactive shell based on our prototypes. Key accomplishments:

1. **Core Scope Definition**
   * Aligned on “AI-assisted DR lesion segmentation and diagnosis, with preliminary report review” as the MVP, and removed non-core modules such as patient management and follow-up.
2. **Project Structure & Routing**
   * Simplified directory to four main views: Dashboard, Diagnosis, Reports, Settings.
   * Updated routes to /dashboard, /diagnosis, /reports (pending/approved/details), and /settings.
   * On login, role check (doctor) redirects into the doctor-only modules.
3. **Dashboard Implementation**
   * Four summary cards: Today’s diagnoses, Pending reports, Flagged cases, Cumulative monthly diagnoses.
   * “Pending Cases” table: lists AI-generated reports awaiting doctor review.
   * “Recent Activity” timeline: shows diagnosis and review events only, omitting follow-up logs.
4. **AI Diagnosis Module**
   * **Upload & Diagnose**: drag-and-drop or click upload; front-end displays segmentation mask, lesion count, lesion types, and confidence, with a one-click “Submit for Review” button.
   * **Diagnosis History**: paginated list of all AI diagnoses, with search and quick-jump to review.
5. **Report Management Module**
   * Two tabs: **Pending Reviews** and **Approved Reports** to manage AI-generated and doctor-reviewed reports.
   * **Report Details** view: shows segmentation overlay, lesion count, and review comments.
6. **System Settings**
   * Static layout for doctor profile and preference settings completed.
7. **Interactions & UX**
   * Built with Vue 3 + Vite + Element Plus + Pinia for a unified, responsive UI.
   * All views use mock data to validate front-end flows and component interactions, easing upcoming API integration.

**Next Steps**

* **Backend Integration**: Finalize and implement API contracts for diagnosis and report services.
* **Model Hookup**: Connect AI segmentation outputs (masks and lesion counts) to the front-end and refine visualization.
* **Review Workflow**: Complete the “Report Review” interaction, including comment input and status updates.
* **UI Refinements**: Polish dashboard cards, tables, and timeline layouts based on real data and user feedback.
* **Testing & Deployment**: Write front-end test cases, prepare a Docker environment, and move toward pre-production validation.