**Biweekly Report**

**Project Title**: *Diagnosis of Diabetic Retinopathy System*  
**Team Member**: Peijin Chen  
**Time Period**: April 7 – April 20, 2025  
**Individual Working Hours**: ~15 hours

**1. Further Literature Review**

Two Kaggle notebooks were studied to support future model implementation and UI integration:

1. **APTOS DR EDA & Starter**: This notebook demonstrates dataset distribution analysis, image preprocessing (resizing, augmentation), and PyTorch data pipeline construction.
2. **CNN for DR Diagnosis (PyTorch)**: A full CNN training workflow is implemented with model structure, training-validation loops, and evaluation metrics such as QWK.

The reading focuses on understanding how data preparation aligns with model accuracy and UI data binding.

**2. Existing System UI Research**

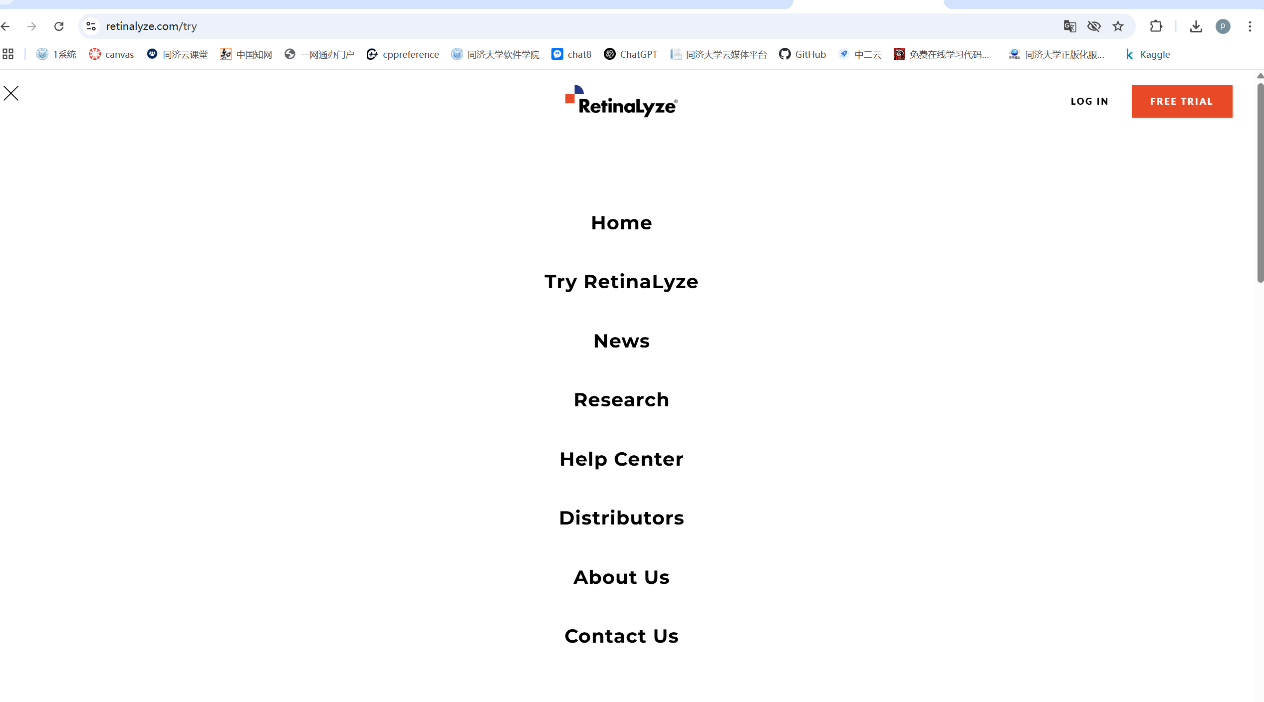
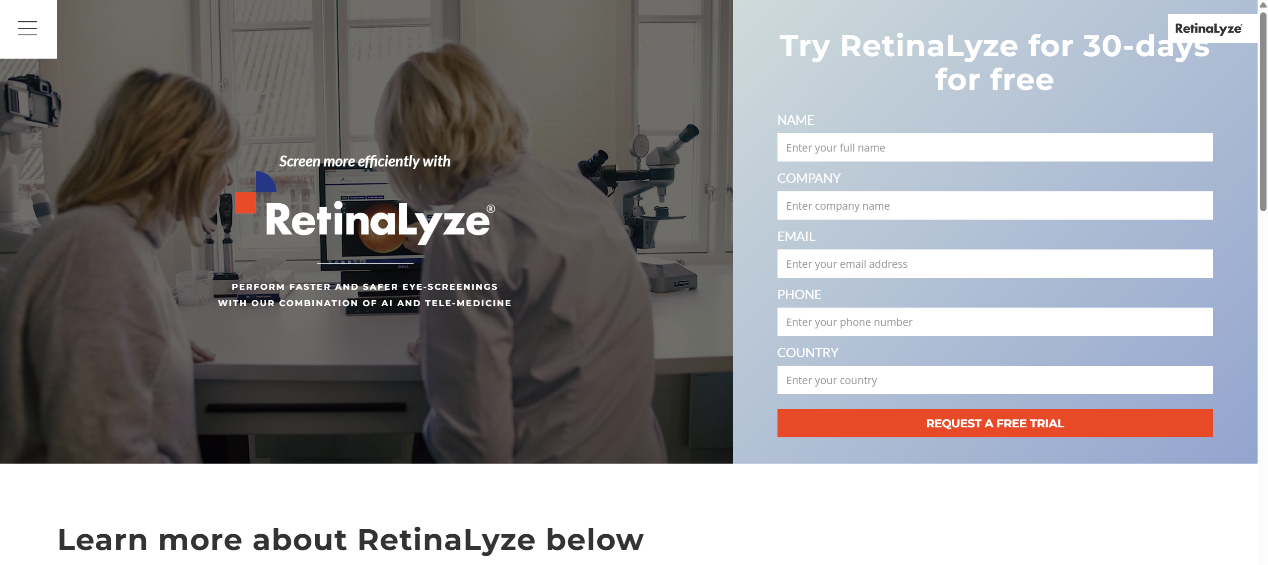
A survey of commercial AI diagnostic platforms was conducted:

* **iCare RETCAD**: Clean clinical layout with detailed result visualization, suitable for doctor workflows

<https://www.icare-world.com/product>.

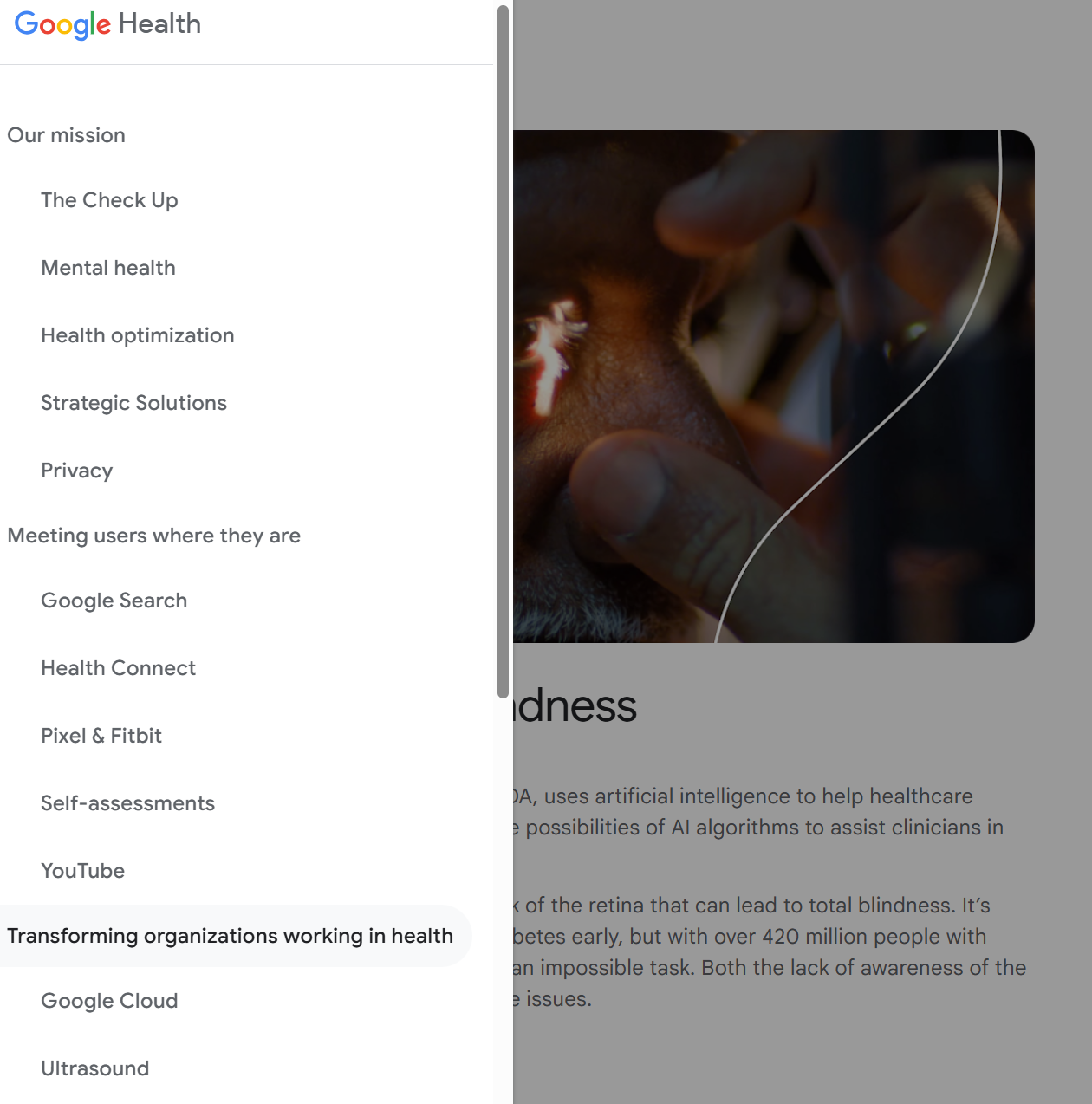
* **RetinaLyze**: Emphasizes fast uploads and simple diagnostics, often integrated into EMR systems.

<https://www.retinalyze.com/>



* **Google Health ARDA**: Mobile-adaptive, minimal interface, ideal for low-resource settings.

<https://health.google/caregivers/arda>



These systems provided key inspiration regarding diagnosis display structure, interface minimalism, and doctor/patient role segregation.

**3. Initial Front-End UI Design**

**3.1 Design Goals**

The interface supports both **doctor** and **patient** roles. Primary features include image upload, diagnosis display, history tracking, and report generation. The visual style adopts a clean white background with blue accents for clarity and comfort.

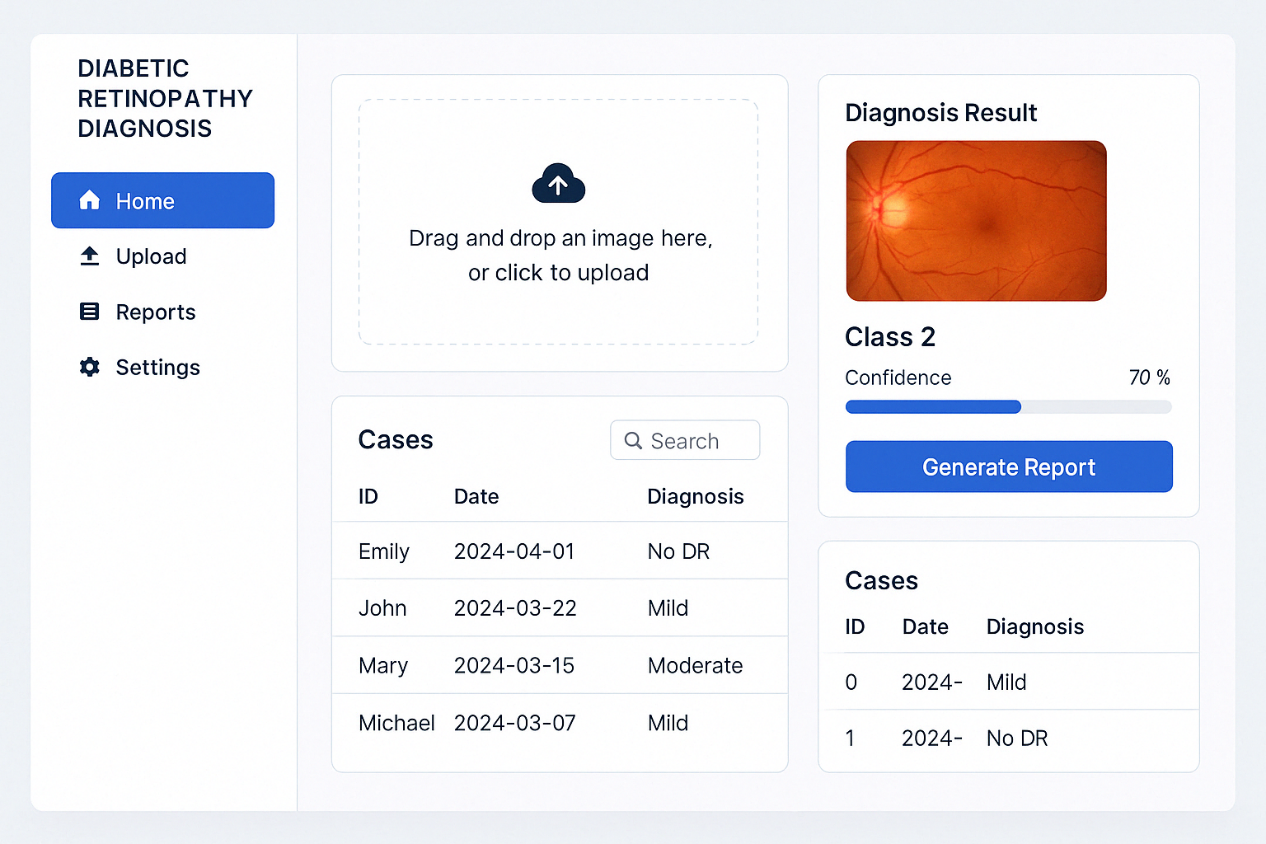
**3.2 Core Functional Modules**

| **Module** | **Function Overview** |
| --- | --- |
| Login/Register | Email-password login; role-based redirection to dashboards |
| Upload Panel | Drag/drop or click upload; supports batch upload and image preview |
| Diagnosis View | Displays uploaded image, diagnosis level, confidence, and heatmap |
| History Records | Displays case list with filtering, sorting, and detail access |
| Report Output | Auto-generates PDF/PNG reports including image and treatment info |

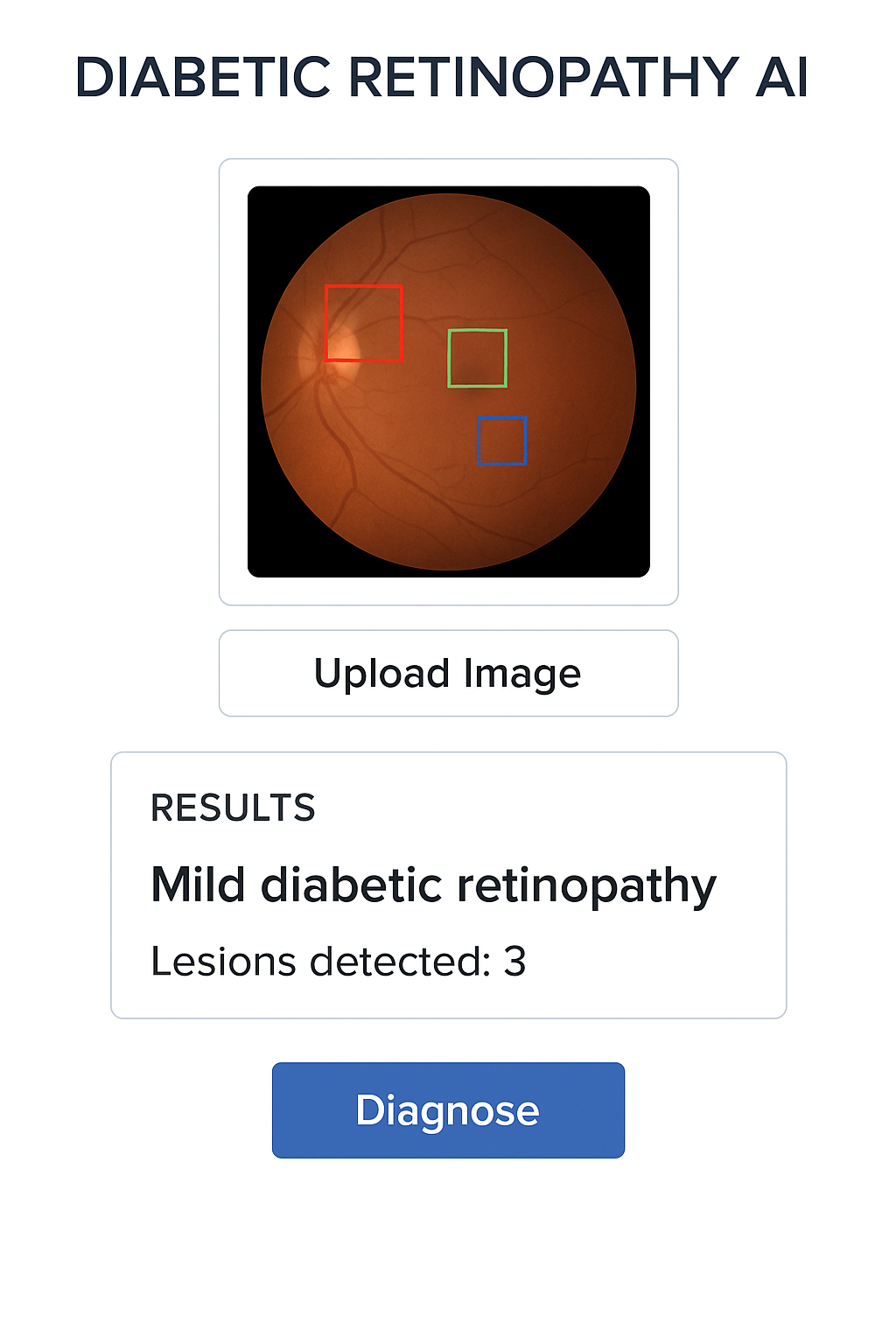
**3.3 Layout Design Structure**

**Main Views**:

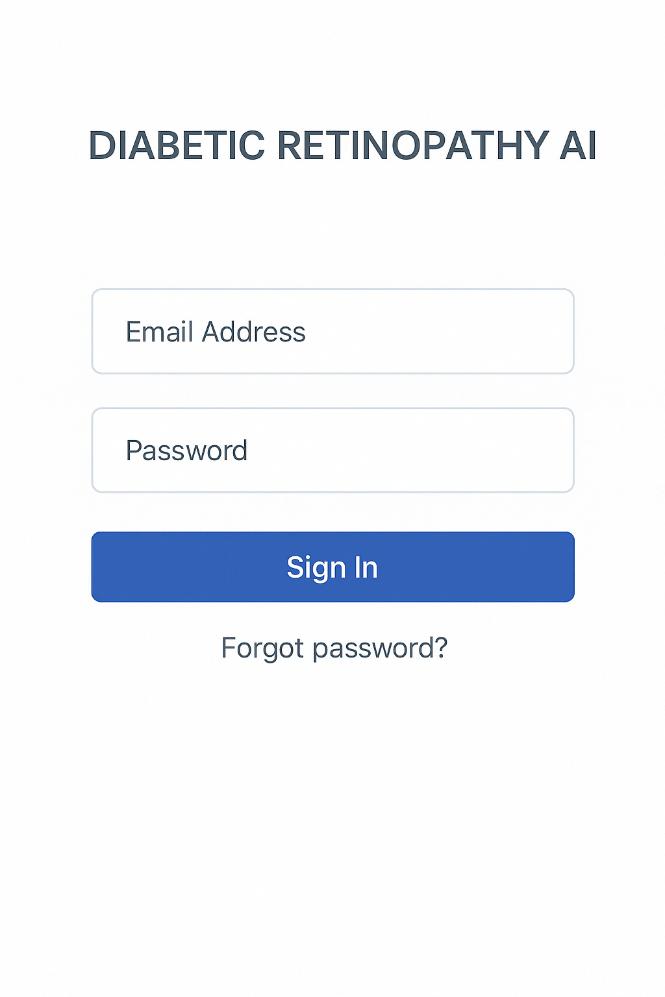
* **Doctor Dashboard**: Sidebar navigation + central image panel + results area + case table



* **Patient Dashboard**: Upload section → diagnostic result → report preview

****

* **Login Page**: Minimal layout with left info panel and right-side login form



**3.4 Technical & Component Plan**

* **Frontend stack**: Vue 3 + Vite + Element Plus
* **State management**: Pinia (user, case data, upload status)
* **Responsiveness**: Adaptive grid layout for mobile and desktop
* **Heatmap rendering**: Canvas-based interaction planned for future integration

**4. Next Steps**

* Implement basic page routing and layout using Vue components
* Complete image upload + preview interaction logic
* Draft and align backend API endpoints
* Design logic for binding model output (grade + image + confidence) to UI layout