Project Report: NeuroLearn

"Hack-A-Thon : AI for Education - AI In Education, Organized by Think+. Personalized AI-driven assessments and practice for students."

Project Name: NeuroLearn
Team Name: Neural Pioneers
- Harshit Kumawat (Team Leader, AI Workflow)
- Koustubh Dave (Database & APIs)
- Himanshu Soni (UI/UX)

GitHub Repository:
https://github.com/Harshit1234G/NeuroLearn
Demo Video:
https://drive.google.com/file/d/1V9CVTGUzRkFH8YoEZyNuWObx
dhW7gIbo/view?usp=sharing
Date of Submission: September 30, 2025

Abstract

Students in the same classroom often get very different scores on the same tests. Traditional assessment methods do not identify the actual reasons behind these performance gaps. Four key learning basics influence student outcomes: listening, grasping, retention, and application. Current evaluation systems miss these individual differences. As a result, they label students simply as "good" or "poor" without providing useful insights. Our Adaptive AI-Powered Learning Assessment and Practice Tool, NeuroLearn, changes assessment difficulty based on student responses. It identifies the specific issues a student has and generates personalized, engaging reports for the student. It also creates professional and insightful reports for teachers and a report for parents on how they can support their children. NeuroLearn offers step-by-step explanations for all the incorrect answers in simple language. It includes a concept primer about the fundamental ideas related to those incorrect answers and organizes a tailored study plan based on the student's diagnosis. We built this with Vite and React, using Express.js APIs and an SQLite database, along with LangChain, LangGraph, and the OpenAI API to ensure smooth user interactions. This kind of educational setup can help students tackle their challenges at their own pace, providing personalized feedback and explanations.

System Architecture

The following is a high level overview of the system architecture.
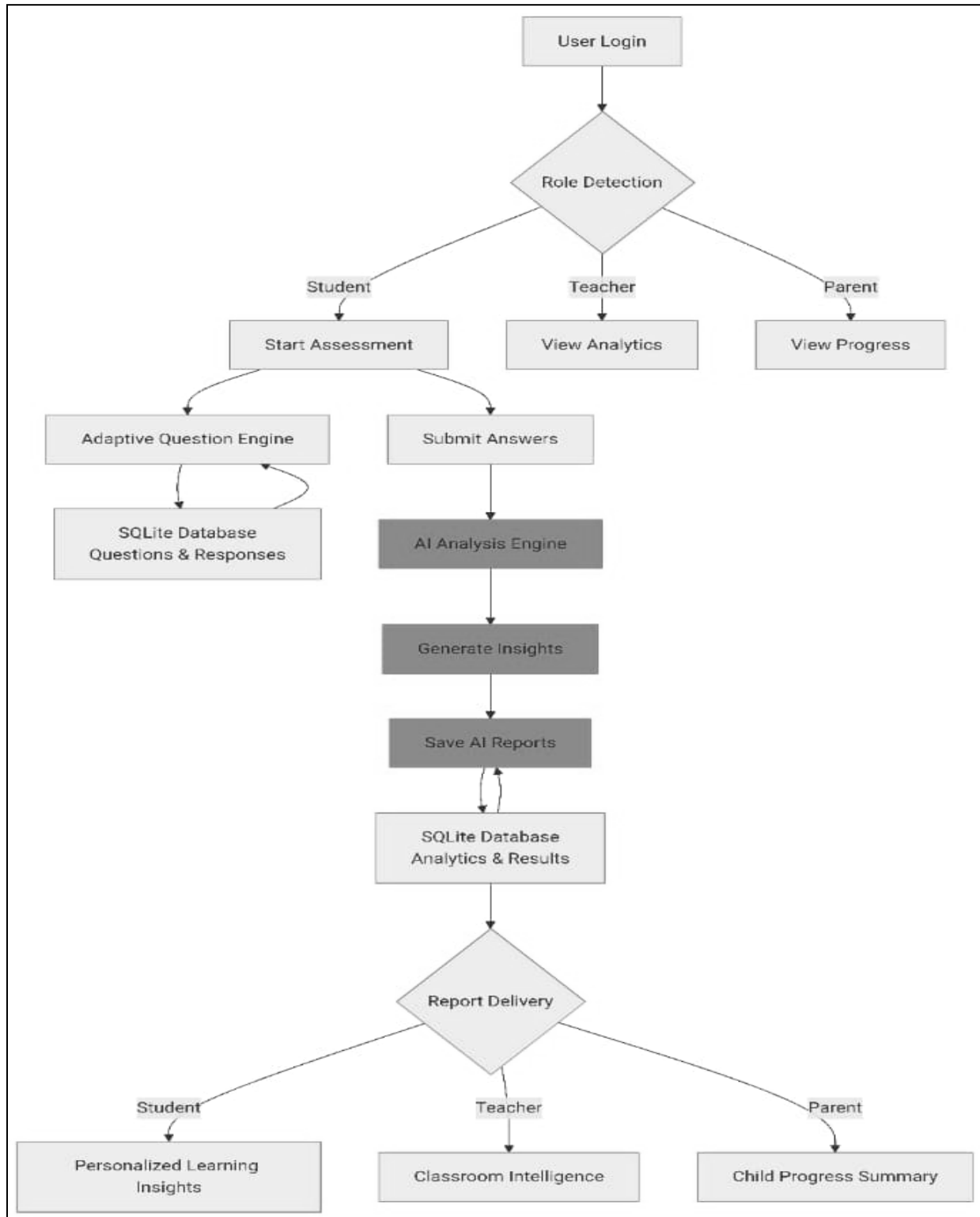
1. Frontend (React + Vite + Tailwind CSS)

The frontend is built with React and Vite and provides a responsive, role-based interface for students, teachers, and parents. Following are the key features:

- **Frontend Framework:** Built with React and Vite for fast development, hot module reloading, and optimized production builds.
- **Styling and Design:** Uses Tailwind CSS for responsive design, ensuring a clean and consistent user interface across devices.
- **Navigation:** Implements React Router for smooth client-side routing between pages and role-based dashboards.
- **Role-based Dashboards:** Offers separate interfaces for students, teachers, and parents, each designed with relevant features and data views.
- **Interactive Assessments:** Includes MCQ-based practice sessions with dynamic question rendering and instant feedback for users.

2. Backend (Express.js APIs)

The backend is built in Express.js and created a RESTful API that manages authentication, assessment, AI analytics, results tracking, and user management. Following are the key features:

- **Authentication and Authorization:** It uses role based access control for students, teachers, and parents. JWT tokens secure the sessions.
- **Assessment Engine:** It supports adaptive assessments with endpoints to start sessions, fetch questions, submit answers, and generate reports.
- **AI Analytics Integration:** Endpoints to store and retrieve AI-generated diagnostics, step-by-step explanations, and personalized study plans.
- **Results and Reporting:** The APIs offer performance dashboards for students, classroom analytics for teachers, and easy-to-understand progress summaries for parents.

3. Agentic AI Workflow (LangChain + LangGraph + OpenAI API)

- **CLI Workflow (Primary):** This is the main end-to-end workflow. It manages assessment, diagnostics, reports generation, and notes explanations directly within the agentic pipeline. This serves as the reference for adaptive assessment logic.
- **Web Workflow (Prototype):** This is a simplified version for the web interface. Here, assessments are managed through REST APIs instead of within the agentic loop. This version is **not fully integrated** into the current prototype, but it can be used for UI integration.

CLI Workflow   Web Workflow
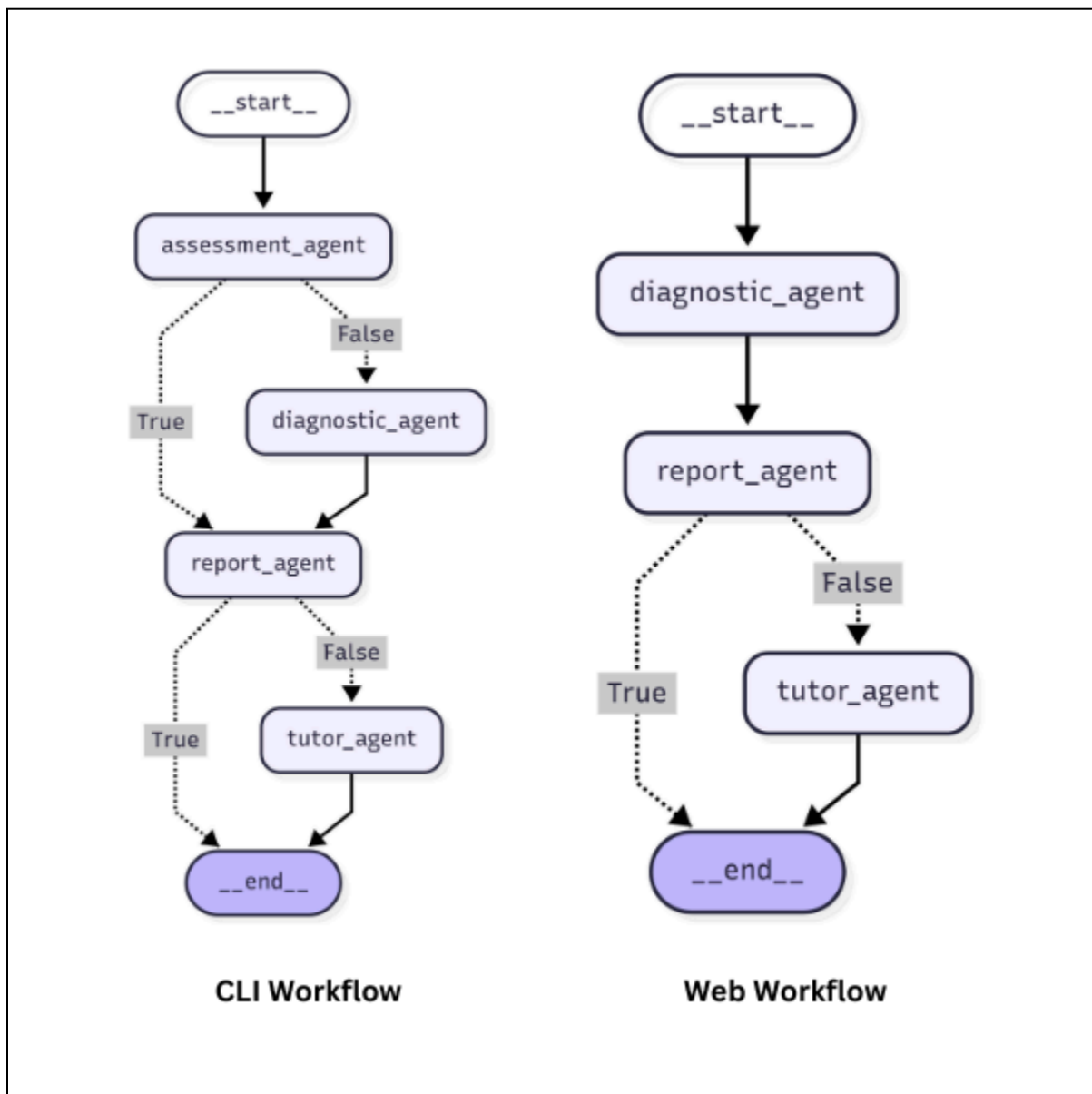
- **Assessment Agent:** conducts adaptive assessments by selecting questions, adjusting difficulty, and tracking student responses, uses gpt-4o-mini.
- **Diagnostic Agent:** analyzes results to identify strengths, weaknesses, and skill gaps, uses gpt-4o-mini.
- **Report Agent:** generates tailored reports for students, teachers, and parents, uses gpt-4o-mini.
- **Tutor Agent:** provides detailed explanations and personalized study guidance, uses gpt-5-mini.
- **Orchestration:** manages the end-to-end workflow, connecting all agents into a cohesive adaptive learning pipeline, using LangGraph.

4. Database (SQLite)

The database uses SQLite3 for light, fast, and portable storage. Here are the key technical details:

Core Tables:
- students: stores user profiles with authentication information.
- parents: linked to students for monitoring progress.
- teachers: stores educator profiles and subject information.
- student_results: records performance metrics and timestamps.
- ai_reports: stores diagnostics, explanations, and personalized recommendations.

Data Relationships:
- One-to-many: students to assessments/results.
- One-to-one: results to AI reports.

Step-by-step simple workflow

1. The student begins the assessment. The system loads the dataset and sets the difficulty.
2. Questions are chosen based on the difficulty level, topic tags, and previous attempts.
3. After each response, the system checks if it's correct and adjusts the difficulty level accordingly.
4. At the end, all answers, scores, and difficulty progress are stored in the student's state.
5. After the assessment, the AI looks at patterns in listening, understanding, remembering, and applying the knowledge.
6. The platform creates outputs: a student-friendly summary, reports for teachers and parents, step-by-step explanations, concept overviews, and a customized study plan.

User Journeys

Sanga (Struggling student)

1. Sanga takes a short assessment and gets several comprehension questions wrong.
2. The system flags "grasping" as the main weakness.
3. Sanga receives a playful game-style report that explains progress and motivates her.
4. She gets clear step-by-step solutions and a short concept primer plus a 2-week study plan with bite-sized exercises.
5. Her teacher receives a professional report with suggested interventions; Sanga follows the plan and retakes a focused mini-test.

Results & Impact

Traditional assessments stop after getting a score. But our platform provides feedback, tailored solutions, and study plans so that students can improve at their own pace and in their own way. For example, Sanga scored 35 out of 100 in a test, but rather than telling Sanga that he failed and he has to improve, our system provides personalized game style reports, explaining all the wrong questions and underlying concepts in detail in an easy to understand and personalized way. For example, the main problem with Sanga is he can not apply the learnt concepts properly, so our system will help him by acknowledging the problem and showing him strategies to overcome this issue. We expect this approach to improve classroom outcomes by turning generic remedial efforts into personalized learning journeys. At scale, such a system could help democratize personalized education in diverse classrooms.

Future Scope

Our current prototype shows that personalized, adaptive assessments with agentic AI are possible. In the future, we see several important improvements:

- **Scalability:** We will move from SQLite to a cloud-based database to handle larger groups of students, and proper asynchronous support.
- **Better AI agents:** Using larger/expensive models, creating more complex architecture, better prompts, more personalization, and adding a chatbot so that students can clarify their doubts.
- **Expanded Features:** We will introduce gamification, support for multiple languages, and improved dashboards for parents and teachers.
- **Proper Integration:** Currently everything is scattered, we submitted a prototype but we want to integrate this into fully functional software.

With these changes, the platform can develop into a strong, scalable solution that supports students, teachers, and parents with truly personalized education.

Conclusion

Traditional assessments often simplify diverse learning challenges into basic scores. As a result, students and educators miss out on valuable insights. Our AI-powered adaptive assessment platform fills this gap by adjusting question difficulty in real time, identifying weaknesses in key learning areas, and providing tailored feedback for students, teachers, and parents. By combining intelligent AI processes with an easy-to-use interface and a dependable backend, the system creates an engaging, data-driven learning environment that meets each learner's needs. Though it is currently just a prototype, this solution could develop into a strong educational tool that promotes inclusion, motivation, and effectiveness in classrooms and beyond.