Authored: A University-Focused Browser IDE

Supporting Academic Integrity in Programming Courses

By: Liam Brown
Advisor: Morgan Brown

Team Members & Advisor

Liam Brown

Solo Contributor

brown5lc@mail.uc.edu

Morgan Brown

Industry Advisor

Project Abstract

Web-Based University IDE

Authored is a web-based programming IDE for university courses, allowing instructors to create coding assignments and students to complete and submit them directly in the browser.

Integrity & Accessibility

The platform includes lightweight activity logging to support academic integrity, along with a simple and accessible interface for both instructors and students.

Streamlined Course Workflow

Authored aims to reduce setup overhead, streamline grading, and provide transparency. It combines a clean frontend with a secure backend for a complete course workflow.

User Stories & System Design

Student

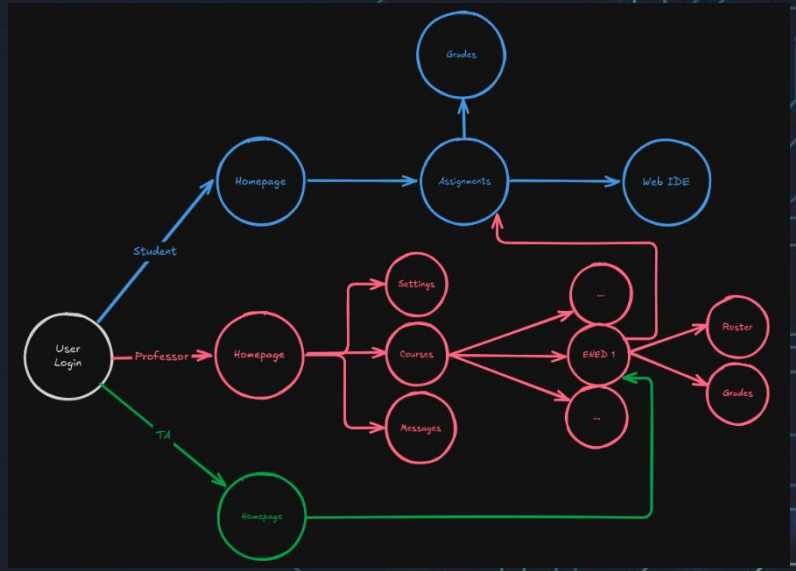
As a student, I want to write and run code in a browser so I don't have to install extra tools.

Instructor

As an instructor, I want to create programming assignments and view submissions in one place.

Instructor

As an instructor, I want to see basic logs of student activity to help verify academic integrity.



Project Constraints

ETHICAL CONSIDERATIONS

Must avoid overly invasive logging; transparency is important.

LEGAL COMPLIANCE (FERPA)

Must handle student data in ways consistent with FERPA.

SUPPORTIVE APPROACH

Intended to support (not police) students.

Current Project Progress

INITIAL ARCHITECTURE

Architecture drafted & core components defined.

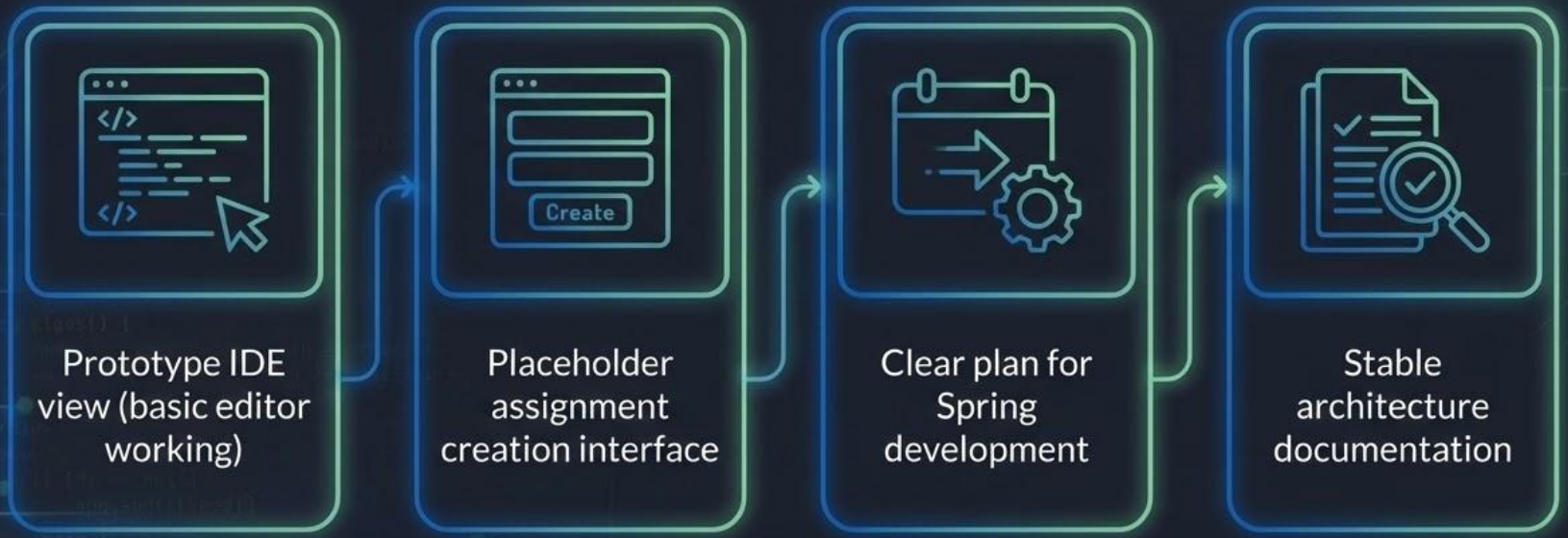
DATA MODEL DESIGN

Courses, users, & assignment models established.

REPO & DOCUMENTATION

Repo initialized; documentation started.

Expected Fall Deliverables



Division of Work

Liam Brown — 100% of development and planning

Frontend editor + UI design

Backend design and APIs

Logging and submission workflow

Testing + documentation

Final demo preparation

Planned Demo for Expo

Instructor creates a coding assignment

Student opens the IDE and writes code

Student runs code in a sandboxed environment

Student submits solution

Instructor views student submission + activity logs

Full workflow: instructor → student → submission review