PROJECT 2 : ALGOLEARN

DESIGN DOCUMENT

CS5610 – WEB DEVELOPMENT

Northeastern University

02/20/2026

1. **Shriya Yarrapureddy Sarath**
2. **Deeksha Manjunatha Bankapur**

PROJECT DESCRIPTION

Overview

AlgoLearn is an interactive educational web application designed to help computer science students master sorting algorithms through visual learning, interactive controls, and gamified assessments. Students watch step-by-step Canvas animations of sorting algorithms (Bubble, Quick, Merge, Heap) with playback controls (play, pause, step-forward, step-backward, speed adjustment), then test understanding through prediction quizzes and multiple-choice challenges. The platform tracks progress, awards achievements for milestones, and features leaderboards to motivate learning. Unlike traditional textbooks or static visualizations, AlgoLearn combines real-time Canvas-based animations with immediate testing and progress tracking to ensure students truly understand algorithm behavior rather than just memorizing steps.

Core Problem

CS students struggle to understand sorting algorithms from textbooks and lectures alone. Traditional learning methods focus on memorization rather than comprehension, leading to poor retention and difficulty applying concepts during technical interviews. Tracing every step of the algorithm and trying to visualize is a cumbersome process.

Solution

AlgoLearn provides an immersive learning experience where students:

- Watch algorithms sort arrays step-by-step with color-coded visual states.
- Control playback speed, step through individual operations, and replay at will.
- Test their understanding through algorithm-specific quizzes.
- Track progress with a comprehensive dashboard showing mastery levels.
- Compete on leaderboards to stay motivated.
- Earn achievements for reaching milestones.

Key Features

1. Visual Algorithm Animations - Canvas-based sorting visualizations for Bubble Sort, Quick Sort, Merge Sort, and Heap Sort

2. Interactive Playback Controls - Play, pause, step forward/backward, speed adjustment (0.5x - 10x)
3. Quiz System - 5 multiple-choice questions per algorithm testing time complexity, space complexity, and algorithm behavior
4. Progress Dashboard - Track completed algorithms, quiz scores, time spent, and learning streaks
5. Gamification - Achievement system with 6 different badges, daily streak tracking, and competitive leaderboards
6. Complexity Analysis - Real-time display of Big O notation, pseudocode, and operation counts.

Target Audience

- Computer Science students (undergrad/grad)
- Self-taught programmers
- Technical interview candidates
- Algorithm enthusiasts

USER PERSONAS

1. **Max - Struggling CS Student**

Age: 20
Major: Computer Science, Sophomore
Location: University campus

Background:
Max is taking Data Structures & Algorithms this semester and struggling to keep up. He finds textbook diagrams confusing and can't visualize how algorithms work in real-time. He has trouble understanding recursion and has failed his last two quizzes on sorting algorithms.

Goals:

- Understand how sorting algorithms actually work step-by-step.
- Pass the midterm exam (25% of final grade).
- Build intuition for algorithm behavior, not just memorize pseudocode.
- Learn at his own pace without classroom pressure.

Pain Points:

- Static textbook diagrams don't show the complete sorting process.
- Lectures move too fast to follow along.
- Can't practice algorithm tracing on his own.
- Doesn't know if he truly understands or just memorized.

How AlgoLearn Helps:

- Visual step-by-step animations he can pause and replay.
- Step-backward feature lets him review confusing parts.
- Quizzes provide immediate feedback on understanding.
- Progress tracking shows which algorithms he's mastered.
- Can learn at 2am when inspiration strikes.

2. **Mike - The Interview Prep Candidate**

Age: 24
Background: Self-taught developer with 2 years experience
Location: Working remotely

Background:

Mike is preparing for technical interviews at FAANG companies. He knows sorting algorithms exist but has never implemented them from scratch. His bootcamp focused on web development, not CS fundamentals. He's been rejected from 3 interviews because he couldn't explain algorithm complexity trade-offs.

Goals:

- Master all common sorting algorithms before next interview.
- Understand time/space complexity deeply enough to discuss trade-offs.
- Build confidence in algorithm knowledge.
- Practice explaining algorithms clearly.

Pain Points:

- Limited CS theory background.
- Doesn't know which algorithms he's memorized vs. truly understood.
- Hard to stay motivated studying alone.
- Needs to learn quickly (interview in 3 weeks).

How AlgoLearn Helps:

- Leaderboard competition keeps him motivated.
- Achievement system provides milestone validation.
- Quiz system reveals knowledge gaps immediately.
- Streak tracker encourages daily practice.
- Complexity analysis helps him understand Big O trade-offs.

USER STORIES

1. **Algorithm Visualization Engine** *(by Deeksha Manjunatha Bankapur)*

    - As a student, I want to select from a library of sorting algorithms categorized by difficulty, so I can learn at my current level.
    - As a student, I want to watch animations to see how the algorithm executes step-by-step.
    - As a student, I want playback controls so I explore algorithm behavior at my own pace.
    - As a student, I want to generate random input arrays of different sizes, so I test algorithm performance with various data patterns.
    - As a student, I want to see pseudocode with the current line highlighted, so I connect visual animation to code logic.
    - As a student, I want to view complexity analysis so I understand algorithm efficiency.

2. **Progress & Challenges** *(by Shriya Yarrapureddy Sarath)*

    - As a student, I want to complete interactive quizzes on time complexity and algorithm behavior, so I test my understanding.
    - As a student, I want to answer prediction challenges about array states after specific steps, so I validate my knowledge with immediate feedback.
    - As a student, I want a dashboard showing completed algorithms, quiz scores, and time spent, so I track progress and stay motivated.
    - As a student, I want to see my learning streak, so I'm encouraged to maintain consistent practice.
    - As a student, I want to earn achievements for milestones (completing all algorithms, perfect scores), so I can feel accomplished by what I have truly mastered.

TECHNOLOGIES USED

Frontend

- Vanilla JavaScript ES6
    - ES6 Modules (import/export)
    - Classes for component organization
    - Async/await for API calls
    - No frameworks (per project requirements)
- HTML5 Canvas API
    - Real-time bar chart rendering
    - Color-coded visual states
    - 60fps smooth animations via requestAnimationFrame
    - Dynamic canvas resizing
- CSS3
    - CSS Grid for responsive layouts
    - Flexbox for component alignment
    - CSS variables for theming (dark mode with pink accents)
    - Smooth transitions and hover effects
- Fetch API
    - RESTful communication with backend
    - JSON data exchange
    - Error handling with fallbacks

Backend

- Node.js - Runtime environment
- Express.js - Web framework
    - RESTful routing
    - JSON middleware
    - CORS enabled
    - Static file serving
- MongoDB Native Driver - Database operations
    - CRUD operations
    - Aggregation pipelines for leaderboard
    - Indexed queries for performance

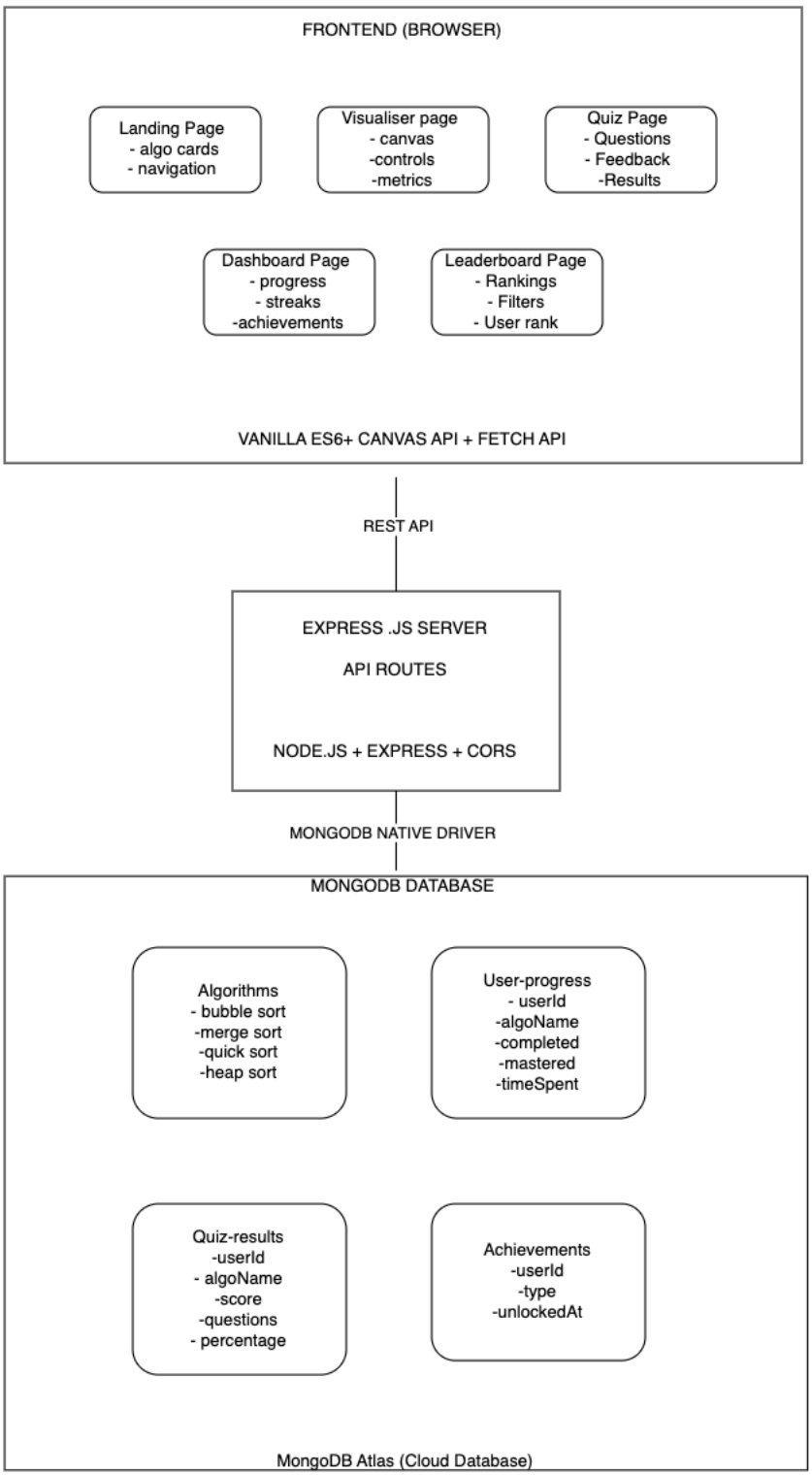
Database

- MongoDB Atlas - Cloud-hosted NoSQL database
    - 4 collections: algorithms, user_progress, quiz_results, achievements
    - Flexible schema for evolving requirements
    - Aggregation for complex queries

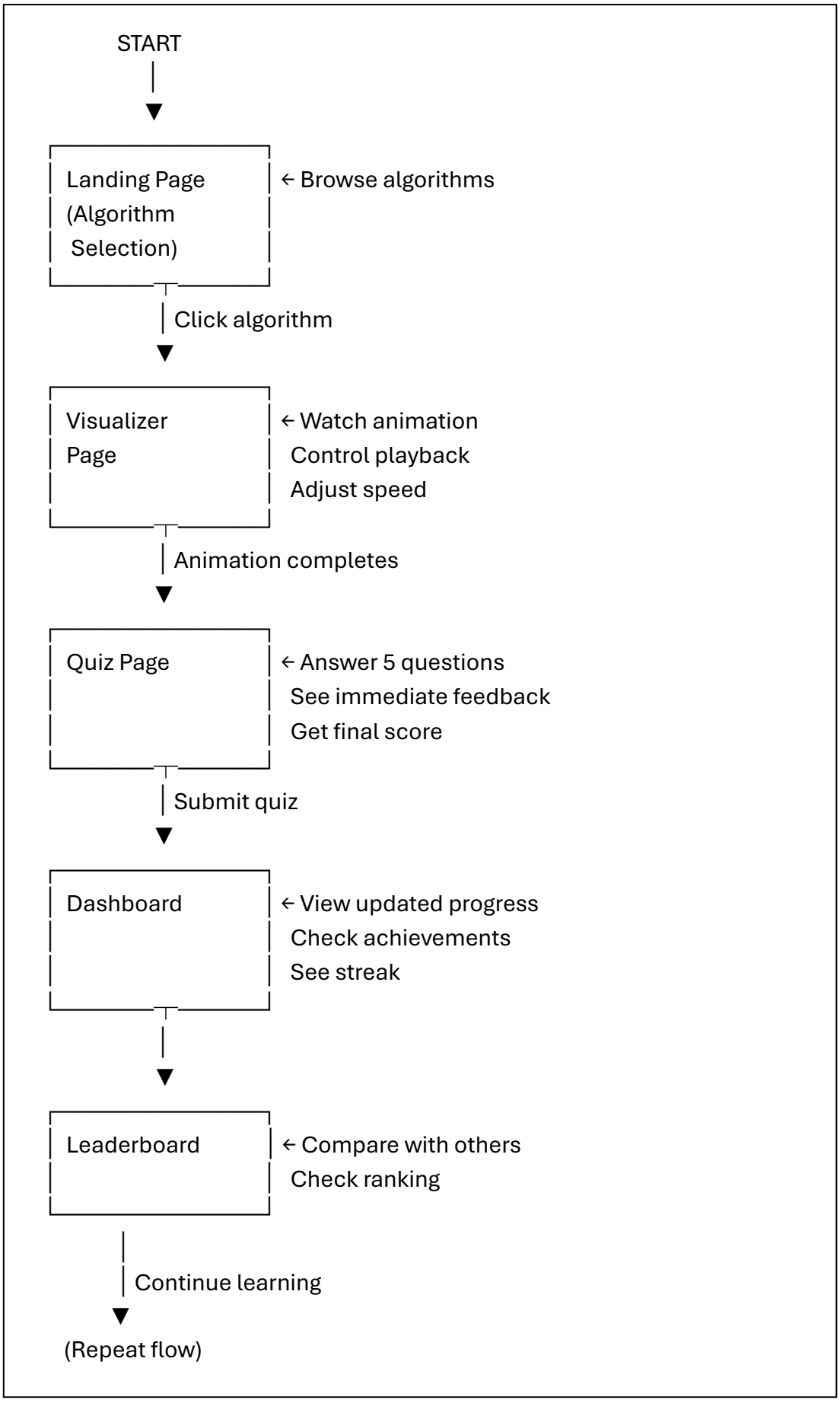
Development Tools

- Git/GitHub - Version control
- VS Code - IDE
- MongoDB Compass - Database GUI
- Chrome DevTools - Debugging

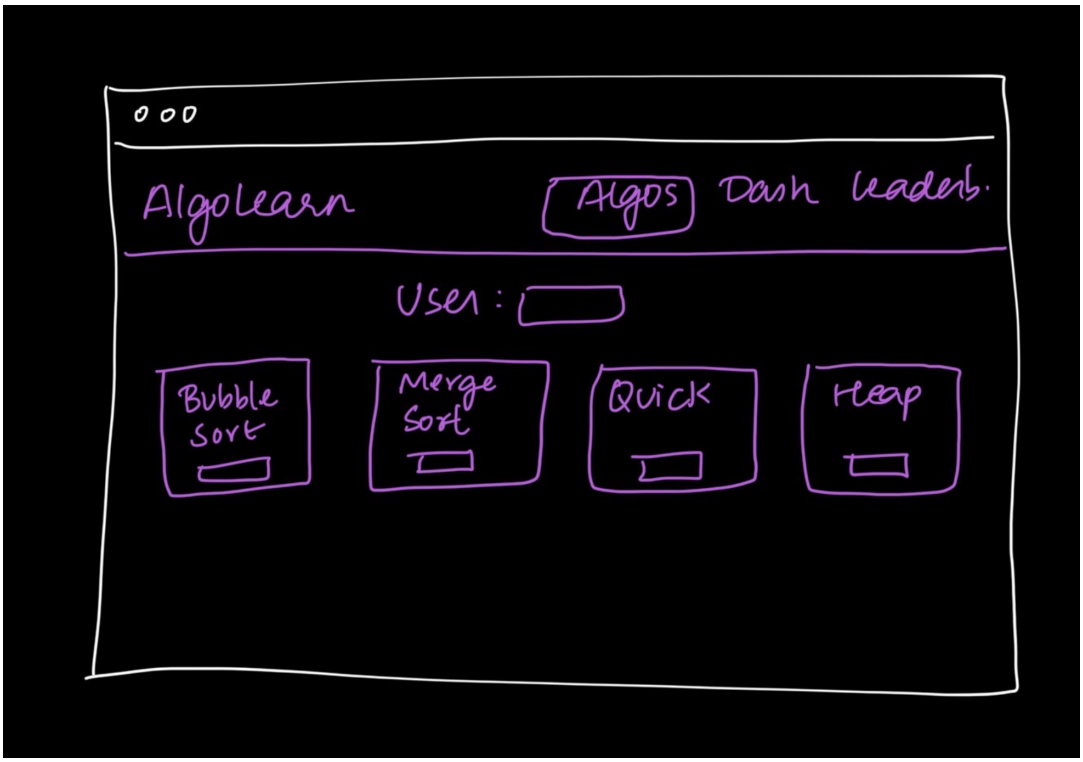
HIGH LEVEL ARCHITECTURE

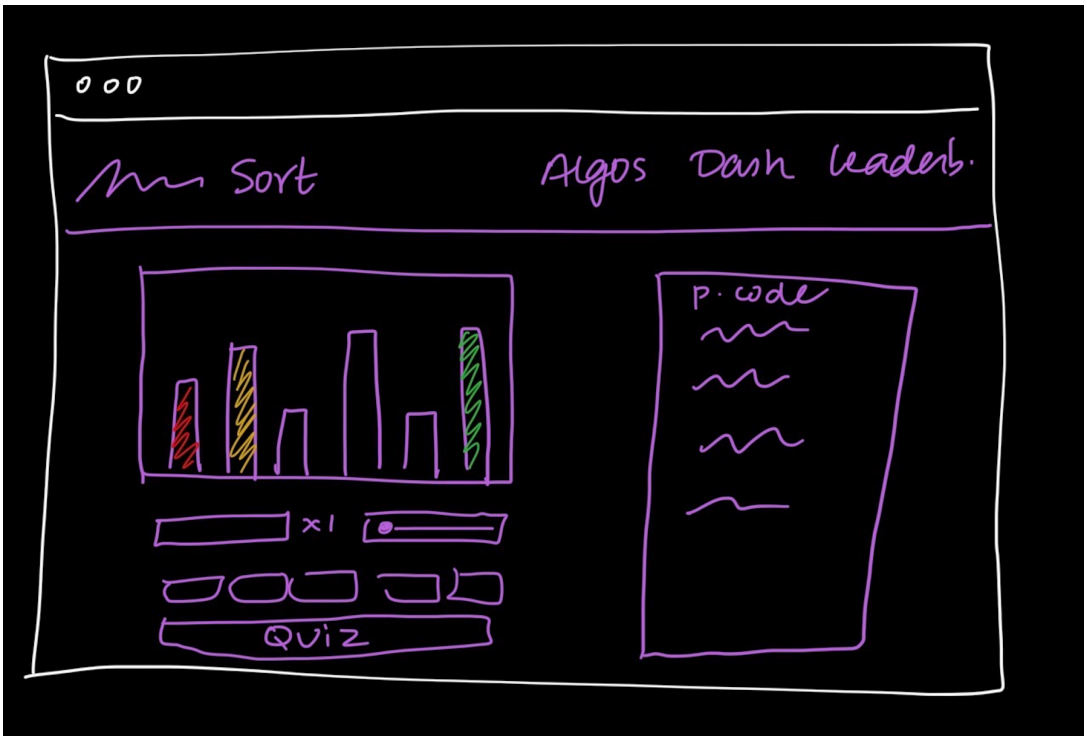


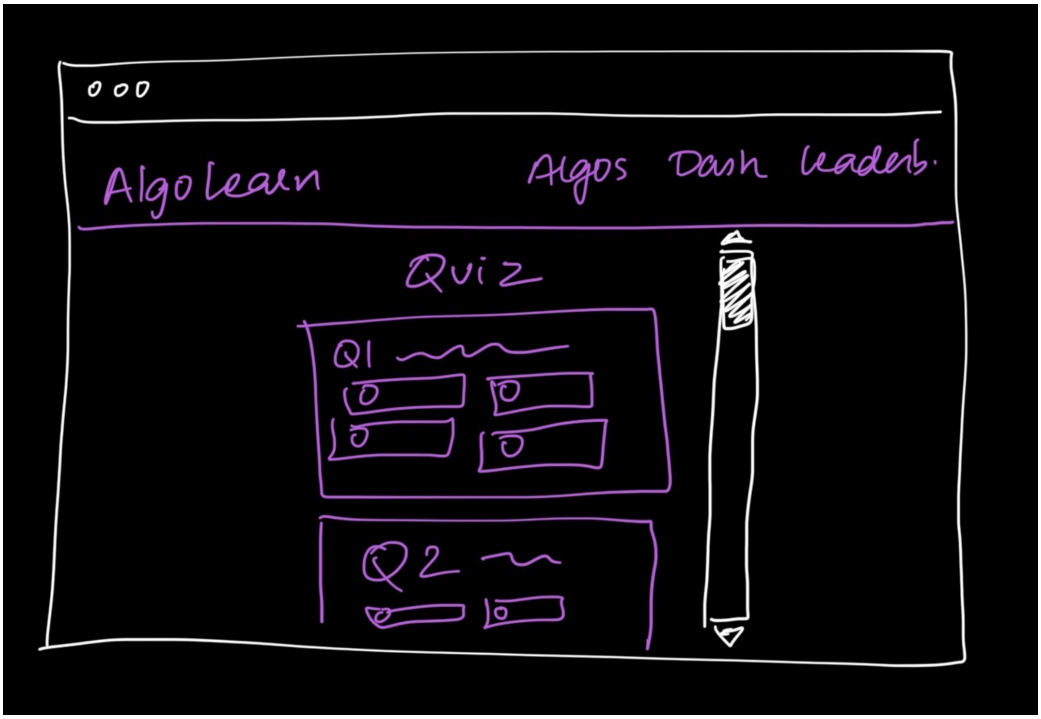
USER FLOW DIAGRAM

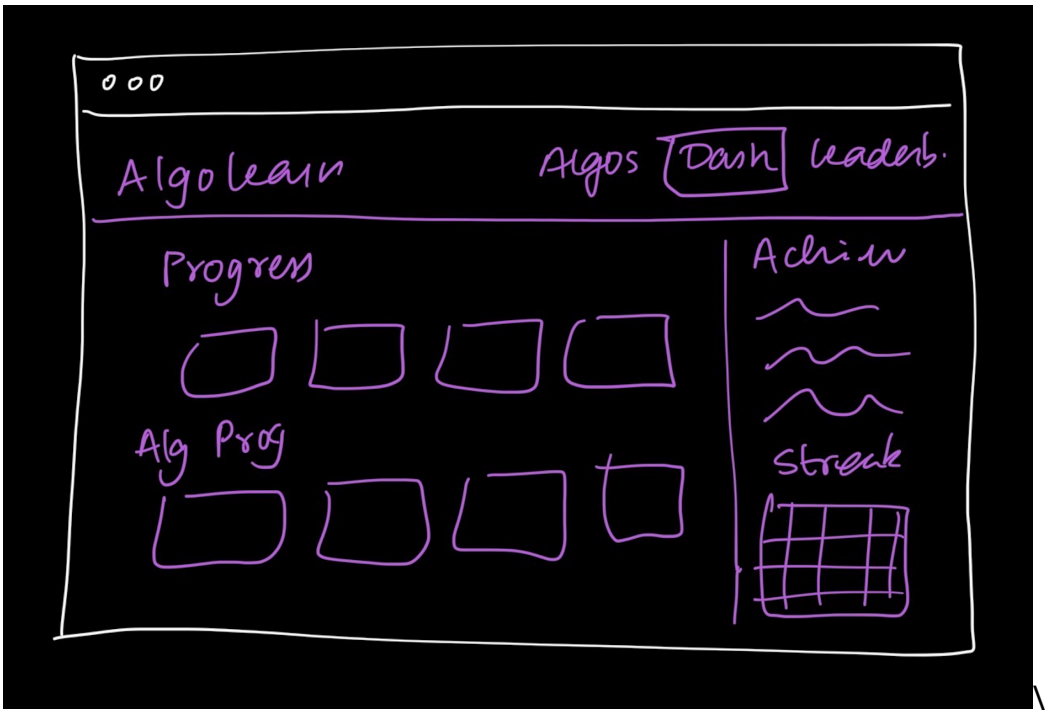


MOCK UPS

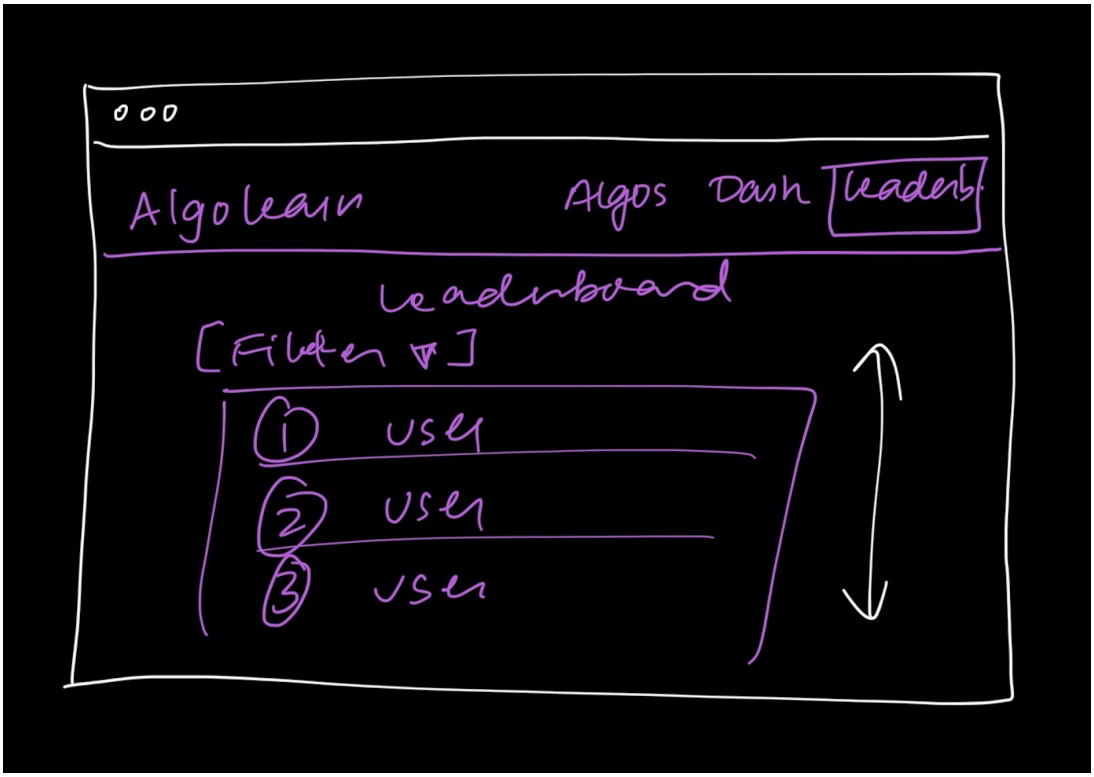

Algolearn
Algos
Dash
Leaderb.
User :
Bubble Sort
Merge Sort
Quick
Heap


Sort
Algos
Dash
Leaderb.
p.code
x1
Quiz


Algolearn
Algos Dash Leaderb.
Quiz
Q1
Q2


Algolearn
Algos Dash Leaderb.
Progress
Alg Prog
Achiev
Streak

\


Algolearn
Algos
Dash
Leaderb
Leaderboard
[Filter ▽]
1 user
2 user
3 user