TEXAS A&M UNIVERSITY
Engineering

Autograder Frontend

CSCE 606
Fall 2024

Autograder Frontend

TEXAS A&M UNIVERSITY
Engineering

Developed using Ruby on Rails and Agile Methodology

Team Members:

- Mainak Sarkar (masarkar@tamu.edu)
- Max Smith (maxsmith271346@tamu.edu)
- Md Hasan Al Banna (mdhasanalbanna@tamu.edu)
- Riddhi Ghate (riddhighate.07@tamu.edu)
- Ryan Gonzalez (gonzalezpear@tamu.edu)
- Saksham Mehta (saksham19@tamu.edu)
- Qinyao Hou (yaoya2618@tamu.edu)

Introduction to the Project

TEXAS A&M UNIVERSITY
Engineering

Overview

The **CSCE 120 Autograder Frontend** is a user-friendly application designed to streamline the creation and maintenance of autograded assignments. It provides an accessible graphical interface for generating assignment GitHub repositories, interactively creating tests, and managing user permissions.

Target Audience

- Course Instructors and Teaching Assistants

Purpose

The application aims to:

- Simplify the assignment creation and constituent test management.
- Enable rapid development and testing of assignments.
- Ensure accessible and efficient management of autograding workflows.

Problem Statement and Solution

TEXAS A&M UNIVERSITY
Engineering

Problem:

- Managing assignments and tests for CSCE 120 is time-consuming and complex.
- Existing processes lack a **user-friendly interface** for creating and maintaining autograded assignments.
- There is no streamlined process for **editing**, **creating**, or **deleting** tests within an assignment.

Solution:

- Provides a **user-friendly frontend** to simplify the creation and maintenance of assignments and tests.
- Streamlines and simplifies the process for **editing**, **creating**, or **deleting** tests within an assignment.
- Offers robust **permission management** for assigning **read/write access** to users in the organization.

UI Overview

TEXAS A&M UNIVERSITY
Engineering

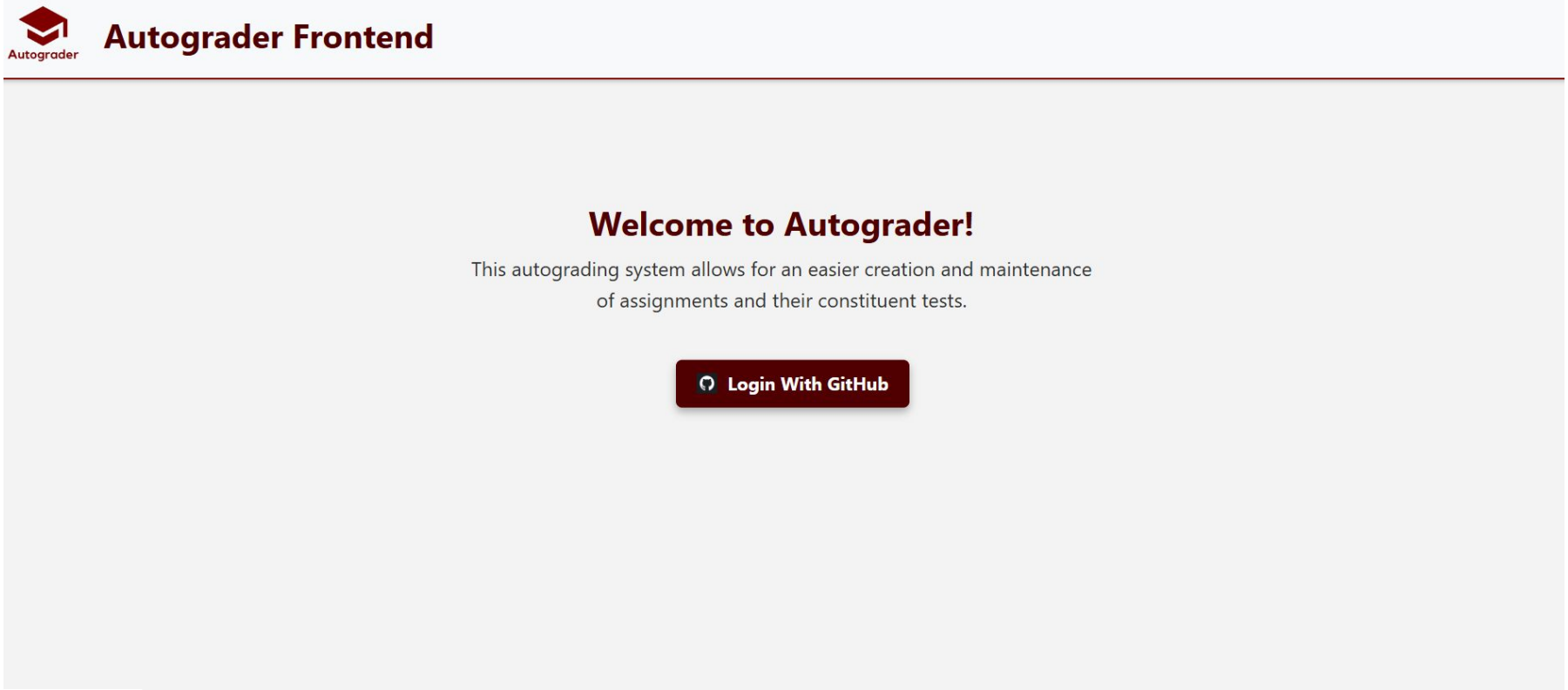
Screenshots of the Application Interface:

The next 4 slides are screenshots of the autograding system's user interface, showing key components and their functionality.

Login Page:

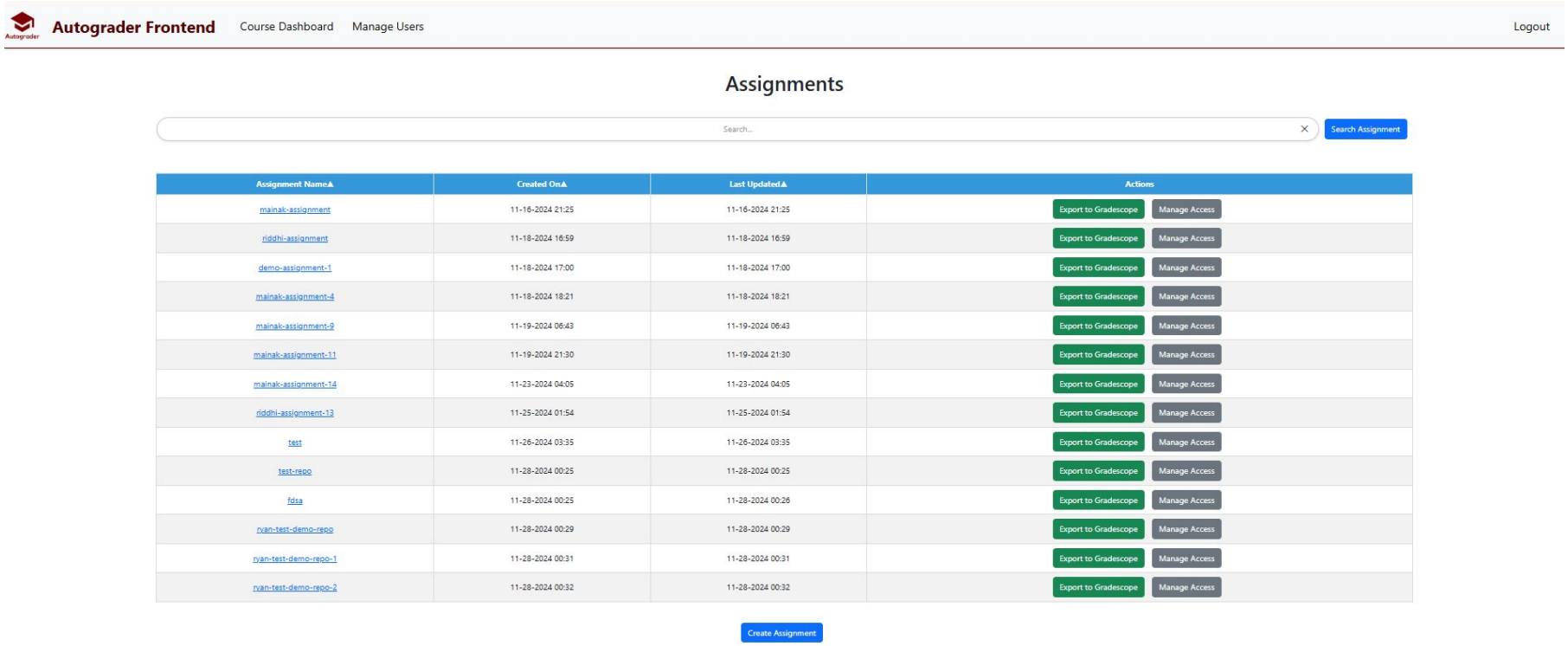
Secure login via GitHub for authenticated access.

TEXAS A&M UNIVERSITY
Engineering



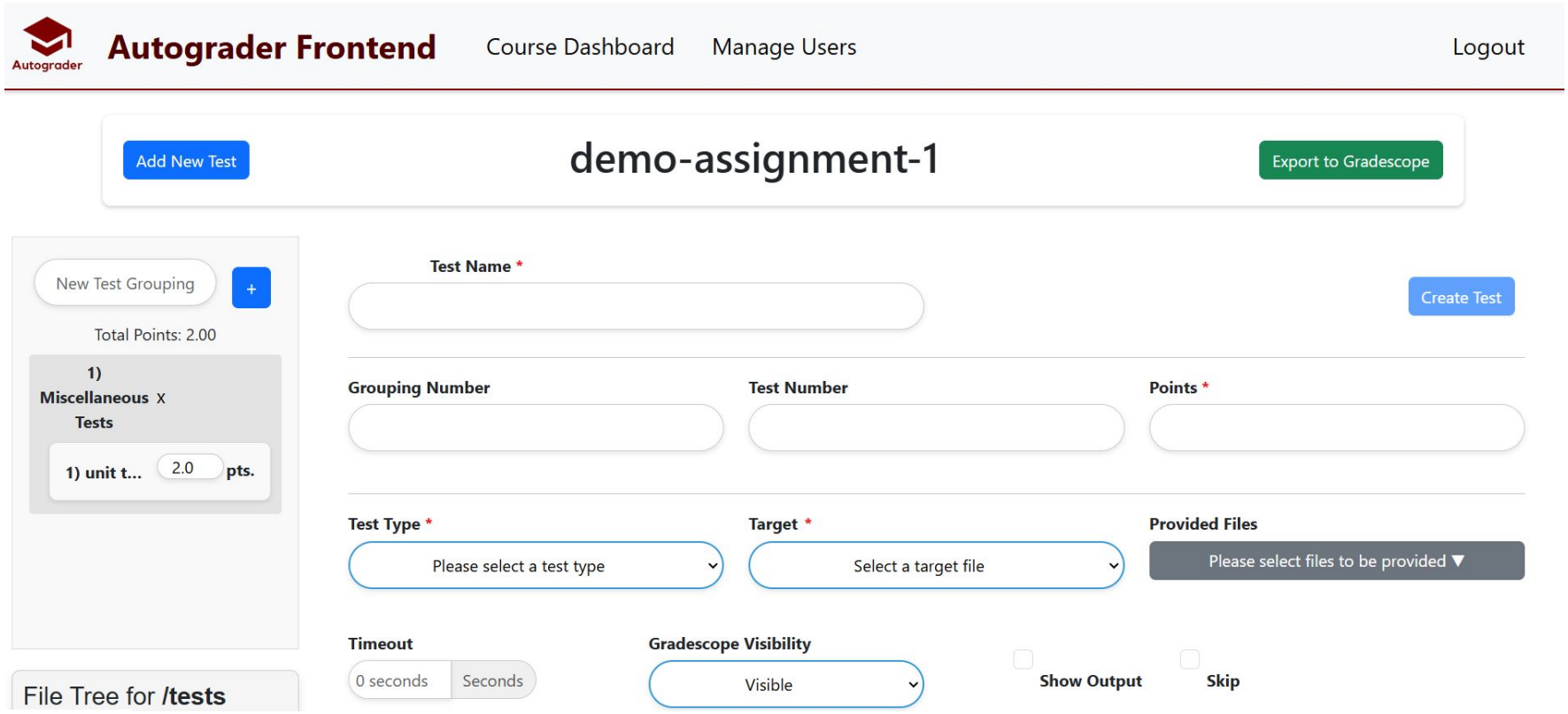
Dashboard:
Central hub for managing assignments, with sortable tables and filters

TEXAS A&M UNIVERSITY
Engineering



Assignment Management:
Interface for creating, editing, and managing tests for each assignment

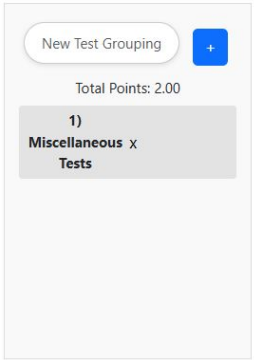
TEXAS A&M UNIVERSITY
Engineering

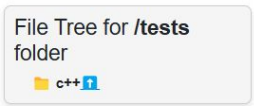


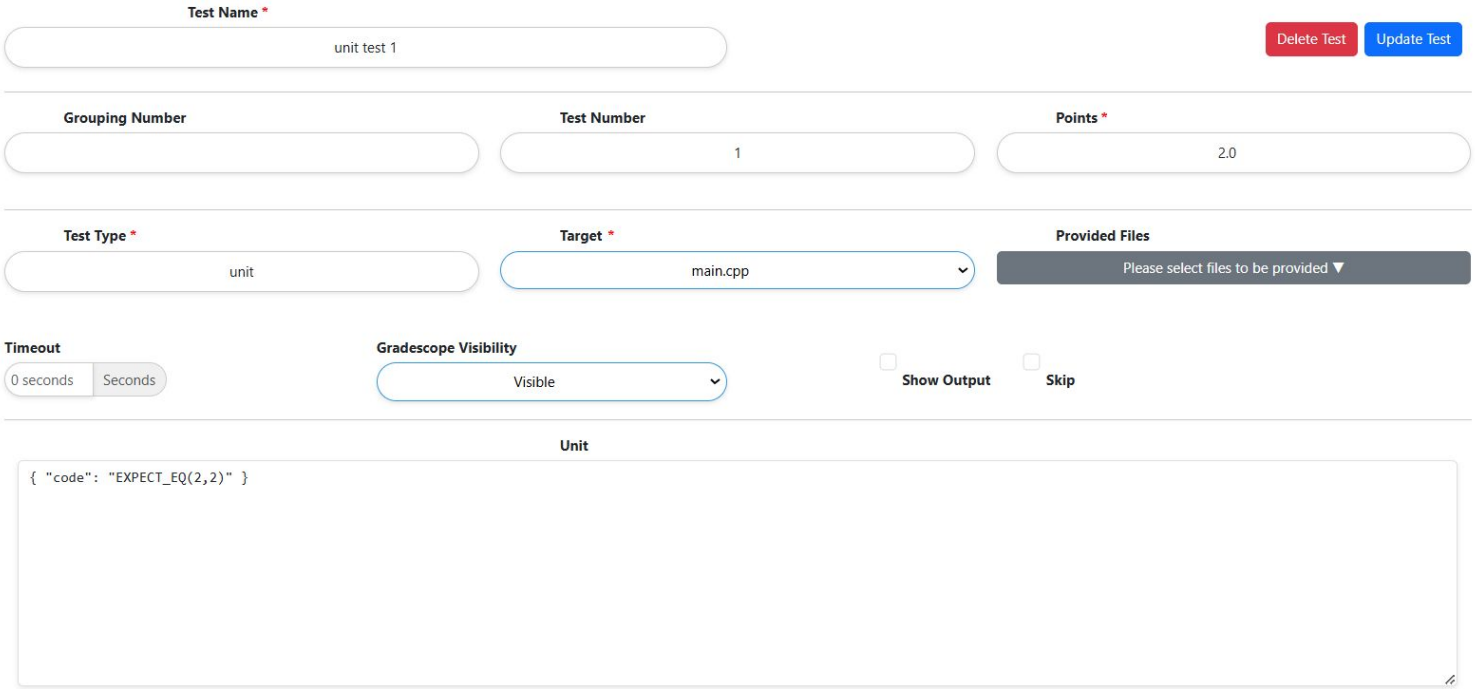
Test Case Management:

Simple, intuitive interface for adding, editing, or deleting tests within assignments.

TEXAS A&M UNIVERSITY
Engineering







Key Features

TEXAS A&M UNIVERSITY
Engineering

User Authentication

- **Description:** Users can log in securely via GitHub, ensuring that only authorized personnel (instructors and TAs) can access the system.
- **Benefit:** Simplifies access control with third-party authentication.

Dashboard Navigation

- **Description:** The dashboard displays all assignments in an organized, sortable table with the ability to filter by name, date, or status.
- **Benefit:** Provides quick and easy access to all assignments for instructors and TAs.

Key Features...

TEXAS A&M UNIVERSITY
Engineering

Manage User Access

- **Description:** Admins can manage user permissions for each assignment, adjusting read/write access as needed.
- **Benefit:** Fine-grained control over who can view or modify assignments.

Assignment Creation and Permissions

- **Description:** New assignments can be created by filling out a simple form, which then automatically generates a GitHub repository under the CSCE 120 organization.
- **Benefit:** Streamlines the process of creating and assigning assignments, with adjustable permissions for TAs and instructors.

Key Features...

TEXAS A&M UNIVERSITY
Engineering

Test Case Management

- **Description:** Create and manage test cases for each assignment, with dynamic fields that change based on test type.
- **Benefit**: Simplifies the process of adding and editing tests, ensuring that tests align with assignment requirements.

Editing and Deleting Tests

- **Description:** Tests can be easily edited or deleted with confirmation prompts to ensure updates are managed safely.
- **Benefit:** Provides flexibility in managing test cases throughout the course.

Key Features...

TEXAS A&M UNIVERSITY
Engineering

Test Grouping and Point Updates

- **Description:** Group tests together and adjust points in real-time. Points for test groupings are dynamically updated as changes are made.
- **Benefit:** Organizes tests effectively and ensures accurate scoring.

Drag-and-Drop Functionality

- **Description:** Tests can be rearranged between test groupings using intuitive drag-and-drop functionality.
- **Benefit:** Simplifies test organization, making it more flexible and user-friendly.

Key Features...

TEXAS A&M UNIVERSITY
Engineering

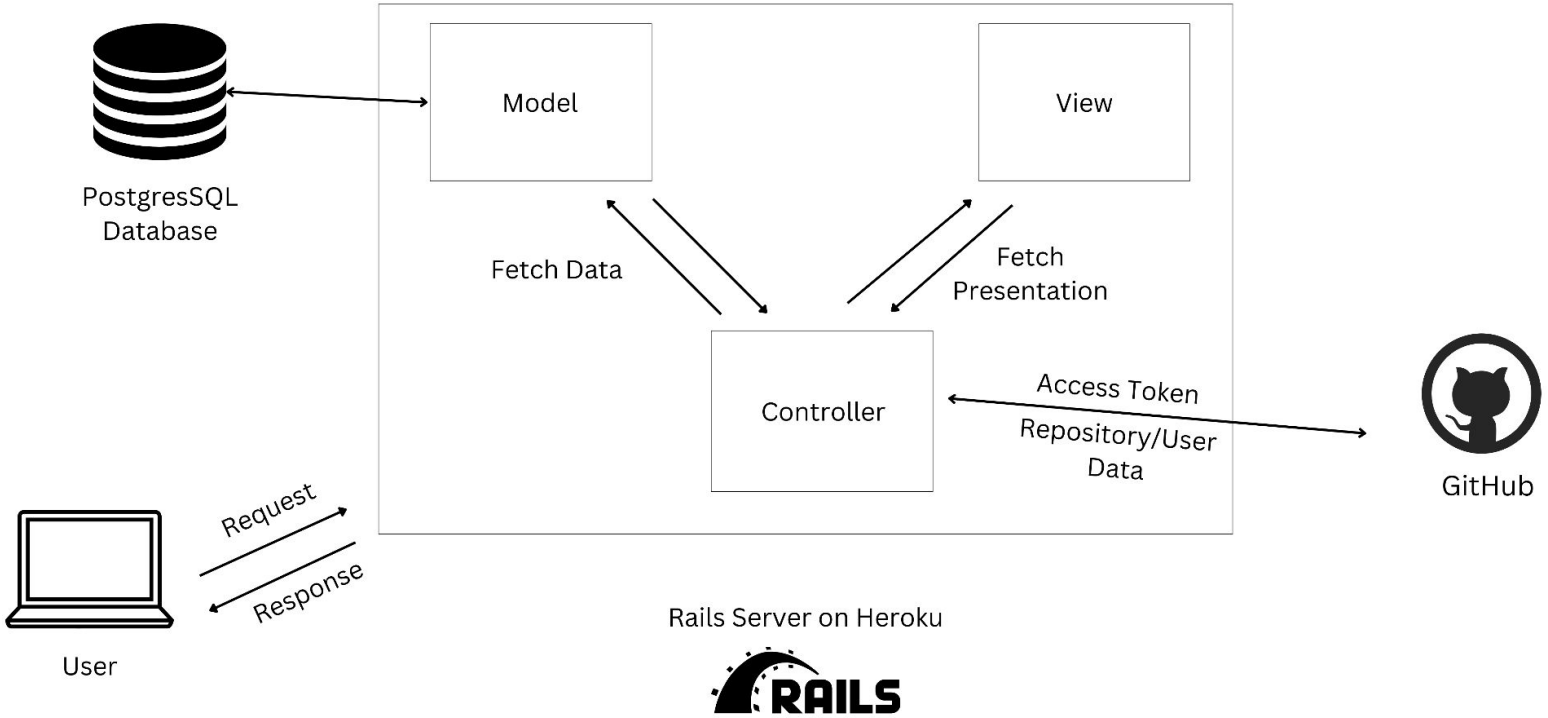
File Management

- **Description:** Files can be uploaded directly to the assignment repository, with an easy-to-use file tree for managing file selections for tests.
- **Benefit:** Ensures that required files for testing are accessible and properly linked to test cases.

Responsive User Interface

- **Description:** The application is designed with a responsive layout that adapts to different devices and screen sizes.
- **Benefit:** Ensures that users have a consistent experience on any device, whether desktop, tablet, or mobile.

Design


PostgresSQL
Database
Model
View
Fetch Data
Fetch
Presentation
Controller
Access Token
Repository/User
Data
GitHub
Request
Response
User
Rails Server on Heroku
RAILS

Challenges

TEXAS A&M UNIVERSITY
Engineering

- **Underestimation of complexity**
  - User stories were often underestimated, leading to last-minute crunches.
  - Dependencies between user stories caused issues with prioritization and execution.
- **Merge Conflicts and Dependency Management**
  - Lack of a priority/order for user stories resulted in complex merge conflicts.
  - Dependencies between tasks were not accounted for during planning.
- **Inconsistent Practices**
  - Adherence to Git best practices was inconsistent, causing workflow inefficiencies.
- **Environment Disparities**
  - Differences between test and production environments led to deployment issues.
- **Ineffective Communication**
  - Initially, the team struggled with communication, leading to confusion and delays.

Lessons Learned

TEXAS A&M UNIVERSITY
Engineering

- **Prioritize Effective Communication:**
  - Communication and collaboration improved over the course of the project, emphasizing the need for open discussions and knowledge sharing.

- **Plan with Dependencies and Complexity in Mind:**
  - Assessing story complexity more realistically and accounting for dependencies is crucial to smooth execution.
  - Assigning priorities and sequencing tasks effectively can prevent bottlenecks.

- **Improve Environment Consistency:**
  - Using a staging branch or environment is essential for continuous deployment practices.

Lessons Learned

TEXAS A&M UNIVERSITY
Engineering

- **Adhere to Best Practices:**
  - Following Git best practices ensures a streamlined development process.
  - Effective testing throughout the sprint prevents late-stage surprises.

- **Focus on Usability and Functionality:**
  - Better wireframes and usability-focused designs contribute to a stronger MVP.
  - Early emphasis on functionality ensures that client requirements are met effectively.

- **Boost Accountability and Task Management:**
  - Breaking down tasks into smaller, manageable subtasks improves task clarity and completion.

Future Scope

- Ability to resolve merge conflicts
- Add/Remove Github user from organization
- Pull remote changes from Github
- Parse tests coming in from remote
- Add an administrator role

TEXAS A&M UNIVERSITY
Engineering

Thank You!