CSCE 606 Software Engineering, Spring 2021
Team Spongebob : Final Project Report

Team Roles

- **Scrum Master:** Hannah Patrick
- **Product Owner:** Nathaniel Raggo
- **Developers:** Mehdi Gorjian , Farnoosh Shirani Bidabadi , Jason Spencer, Alireza Safdari

Project Links

- Pivotal Tracker
- GitHub repository
- Heroku Deployment
- Videos: Poster, Demo, and Combined

Project Overview

As part of the State of Texas Assessments of Academic Readiness (STAAR) program, Texas schools are evaluated based upon students' performances, progressions, and improvements in math, reading, and writing. As a result, schools are interested to predict how students will perform on their next STAAR Test. Educational Leadership Research Center (ELRC), our stakeholder, helps the schools by developing machine learning models to predict student's scores. We developed an intuitive website for ELRC that allows teachers and schools to use the AI model to predict their students' score. Currently, our website predicts Grade 5 Math scores, with more models being added in the future.

The user uploads a CSV file containing student information and their scores from previous STAAR tests. Then, the AI model predicts student STAAR test scores and outputs them into a CSV file. This file can be downloaded by the user on the page that appears after the upload. ELRC was concerned with the website's ease of use for teachers and schools. We addressed their concerns by providing an intuitive user interface, consistent styling, and limiting the number of pages available to the user.

Scrum Iterations

- **Iteration 0** *(0 points)* - Project setup, develop initial user stories with lo-fi UIs, explore legacy code
  - Customer Meeting: Monday April 12th, 2021 at 5PM CST
  - During our first meeting, we met the clients and they introduced the legacy project we would be working on.

- **Iteration 1** *(4 points)* - Additional exploration of legacy code, initial testing, fixes to log-in functionality, addition of file upload functionality
    - <u>Customer Meeting:</u> Monday, April 19th, 2021 at 5 PM CST
    - We demoed user registration, user login, and user file upload functionality. The clients were pleased with our progress.
- **Iteration 2** *(3 points)* - Overhaul and restructuring of legacy code, more testing, styling for the header, home, and about pages
    - <u>Customer Meeting:</u> Friday, April 23rd, 2021 at 5 PM CST
    - We demoed the changes to the home and about pages and the addition of the logo in the site header. We briefly discussed the corrections to the backend and database. The clients remarked that the project was coming along well.
- **Iteration 3** *(6 points)* - Full integration with AI model, additional styling
    - <u>Customer Meeting:</u> Friday, April 30th, 2021 at 5 PM CST
    - We did a full demo of the application, most importantly showing how a user can upload and download a CSV file containing the AI's STAAR test predictions. The customers stated that we did an "awesome job"!

Legacy Code Exploration

We spent time familiarizing ourselves with the existing legacy application by reading through the code, deploying it to Heroku to interact with it, and using tutorials for Python and Django found online. We initially believed it would be sufficient to update the existing code to add new features. However, at the end of Iteration 1 we made the decision to refactor the entire application and its structure due to the poor architecture and needless complexity of the existing code. Our new application is divided based upon functionality and has a properly structured database. In contrast to the legacy application, we did not implement our own authentication, rather opting to use Django's allauth package which provides everything needed for account management and overall simplifies the code.

Testing

In the timeframe we were given, it was not feasible to successfully implement BDD/TDD. This is because learning how to use test suites is not a quick process, and we were given a short period to deliver our application. In addition, due to most of the team's inexperience with Django, writing tests for unfamiliar code that does not exist is a significant challenge. Instead, we used our tests with Github's CI to verify that new features did not cause the application to deviate from its expected behavior.

User Stories

Complete

Refactored Legacy Stories

- ➔ As a teacher, I want to be able to log in, so that I can access the application under my profile *(2 points)*
- ➔ As a teacher, I want my credentials to be remembered after I sign up, so that I can continue to use my same login information *(2 points)*

Refactoring the previous legacy implementation of the account functionality took place in two stages. First, we replaced the custom authentication code with Django's allauth package as described above. Second, we connected our application's database to Heroku so the credentials of the user would be saved even after the page was redeployed.

Other Legacy Stories

- ➔ As a teacher, I want to be able to upload and download a CSV from the webpage, so that I can interact with the AI application *(4 points total)*: We implemented the file upload functionality first for 2 points. Then, we added downloading a CSV file and pages for both functions to allow for the user to interact with the AI model for 2 points. The website only allows CSV files to be uploaded.
- ➔ As a teacher, I want my file to be used by the AI prediction, so that I can receive predicted STAAR test scores *(2 points)*: A trained AI model from team ThinkSmart for Grade 5 Math scores was successfully integrated with our website.

New Stories

- ➔ As a stakeholder, I want the Home and About pages to be styled, so that they will better represent the project *(1 point)*: The website displays the ELRC logo on a maroon header, a place for text on the Home and About pages, and a school-themed background.
- ➔ As a stakeholder, I want the Home and About pages to contain our information, so that it will represent and explain the project *(1 point)*: The Home page explains ELRC's goal with this project and instructions for the site. The About page contains ELRC's history and contact information.

- As a teacher, I want to be able to download a CSV template, so that I can properly format my data for the AI prediction *(1 point)*: The website provides a download link for the CSV template on the “Upload File” page.

Bugs & Chores (0 points)

- Develop and write tests.
- Restructured the backend of the application to increase functionality.
- Fix code styling issues from Deepsource.io integration.
- Redirect to “Upload File” page after user login.
- Implemented consistent styling across all pages.
- Disable unnecessary account avatar generation. Saves ~10kB/account.

Incomplete

- As a teacher, I want the website to have multiple AI models, so that I can get STAAR test predictions for all subjects and grades *(3 points)*: Due to time constraints, we were not able to get every model from team ThinkSmart. We were able to add fields on the “Upload File” page for users to select the Grade and Subject for their file so in the future the correct AI model is selected.
- As a teacher, I want to be able to reset my password, so that I can log back in if I forget it *(2 points)*: Allauth provides the structure for this functionality but needs a connected mail service to send the password reset link. We were not able to get this set up due to time constraints.
- As a stakeholder, I want the webpage to use Texas A&M branding, so that it aligns with Texas A&M *(1 point)*: We prioritized implementing the AI model and basic styling over this feature.
- Automated Way to Delete Files *(0 points)*: Since there is not a way for users to re-access the original uploaded files, there should be a way of removing them from the database.

Configuration Management

We used a requirements text file to ensure installed Python packages were consistent across all of our environments. Additionally, we took advantage of Github’s CI functionality to ensure any pushes to main passed existing tests. Also, in addition to unit and acceptance testing, we wrote tests that checked the value of certain configurations and settings within our code. Github’s CI workflow is set up to run for all pull request creations and shows an error when a test fails.

We did not do any spikes. We had many branches, however most of them were just used to complete a story and do not hold significance on their own. The important branches are:

- **main:** our current application
- **main_OLD:** previous application before our restructure
- **iteration0, iteration1, iteration2, iteration3:** contains the application code present at the time of each iteration and the corresponding documentation for submission

We have a total of three official releases for each iteration we wrote code: Iteration1, Iteration2, and Iteration3 (or final).

Tools

Name	Description	Benefits
Github CI	Define workflows to perform automated tasks on code	Inside Github. Run tests automatically on pull request.
Deepsource.io	Perform static analysis on code to find styling issues, anti-patterns, bug risks, etc.	Integrates with Github. Check code quality concerns automatically on pull request.
GitUML	Generate UML diagram for all or selected parts of Github repository	Integrates with Github. Simplifies diagram creation process via automation.
Django's Allauth	Set of applications for different authentication types and account management	Simplifies application by omitting the need to write and test authentication code.
Python's Coverage	Generates test coverage in an HTML format	Provides test coverage in easy to read format.
Django's Test Tools	Suite of tools for testing code in Django	Simple and versatile: can be used to write a variety of tests, including unit and acceptance.

Issues

A major issue we faced when deploying our application to Heroku was integrating the database. It was discovered that our application was not connecting to the Heroku database in the first place. It required extensive troubleshooting to learn that we needed to manually run migrations in Heroku's CLI in order to connect the Heroku database. Besides this problem, we had no other issues with the deployment of our application. Also, we did not face any issues using AWS Cloud9, Github, or Pivotal Tracker.