



# Scribe Scholars

PREPARED BY Team 7

Riley Robertson, Jake Koontz, Walter Jacquette, Kyle Brown, Jeremy Putnam, and Eric Thiem

## What went well?

Overall, sprint one came together very well. All the main features that we wanted to get accomplished were completed. Being that five of the six of our team had no prior web development experience, the first two weeks were quite slow in terms of progress. We all made good progress on our own tasks in the beginning but nothing connected together until the last week. All the main features that we wanted to get done were accomplished minus some design decisions we changed through creation that we will talk about below.

### User Story #:

#### User Story

- ☐ Task
- ☐ Comments / Discussion

### User Story #1:

#### As a user, I would like to be able to create an account

- ☐ Implement responsive UI for create account page.
  - ☐ Create account page looks good on all devices and screen sizes
- ☐ Connect user login information to Firebase Authentication.
  - ☐ When user enters their data into the fields and hits submit, the data is saved into Firebase authentication
- ☐ Add extra user data to Firebase Database.
  - ☐ We decided to do it all on one page so user inputs all data at once when creating an account.
- ☐ Test to ensure the form rejects malformed information.
  - ☐ When the user inputs bad email, password, or leaves any fields blank, an error message will appear asking them to fix it.

### User Story #2:

#### As a user, I would like to be able to login to my account.

- ☐ Implement responsive UI for the login page
  - ☐ Login page looks good on all devices and screen sizes.
- ☐ Set up request to Firebase to check for a user in Firebase Authentication.
  - ☐ The request is sent to firebase and then firebase responds with use authentication data.
- ☐ Test to ensure the form rejects users that are not found in Firebase Authentication and prompts them to try again
  - ☐ When a user inputs any form of bad input from bad user name or wrong password they are prompted with an error message.

### User Story 3:

#### As a user, I would like to be able to reset my password.

- ☐ Set up Firebase to send password reset emails
  - ☐ When user navigates to settings page they have a button to send them an email with password reset instructions. (Talked about in what went wrong)

- ❑ Display a responsive page where users can input a new password.
  - ❑ After user clicks the reset link in the email they are prompted to input a new password that looks good on all devices
- ❑ Save new password to Firebase Authentication
  - ❑ Once user submits new password the Firebase authentication is fixed with the new authentication and deletes the old one.
- ❑ Test to ensure old password no longer works for user.
  - ❑ Manually tested changing a password and trying the old one and it does reject it.

#### User Story #4:

##### **As a user, I would like to be able to adjust account information.**

- ❑ Implement a responsive main settings page..
  - ❑ Settings page looks good on all devices and screen sizes.
- ❑ Allow users to change all personal information and sign out.
  - ❑ When users navigate to personal settings they have the ability to change their email, phone, and description.
- ❑ Allow users to navigate to classes enrolled in.
  - ❑ On the side bar there is a tab to switch to classroom settings
- ❑ Test to ensure information is synced with Firebase Database when users edit any field in settings page.
  - ❑ Manually tested changing user data and ensuring that firebase is updated accordingly.

#### User Story #5:

##### **As a student, I would like to be able to see an overview of all my classes**

- ❑ Implement a section in the settings tab which will show their current classes.
  - ❑ The settings page has a list of all classes the student is enrolled in, and the list updates immediately after joining or leaving a class.
- ❑ On the class page, there will be an appealing presentation of all classes currently enrolled.
  - ❑ All classes are listed in an appealing font on the sidebar of the home page.
- ❑ Implement a button to start adding a class.
  - ❑ The user is able to enter the code to join a class, and alerts are given if the code does not exist or is not valid.
- ❑ Implement a delete button to remove a class.
  - ❑ When a user clicks the trash symbol, the class is removed from their screen and the database is updated.
- ❑ Test to ensure that classes are altered properly.
  - ❑ Cloud Firestore is updated immediately after an altering, including joining a class, creating a class, leaving a class, and updating personal information.

#### User Story #6:

##### **As a student, I would like to be able to have an easy to navigate landing page**

- ❑ Implement a sidebar containing student's classes.
  - ❑ The sidebar holds all the student's classes, as well as a link to the settings page and the home page.
- ❑ Implement a calendar in the middle of the page with upcoming deadlines.
  - ❑ The calendar has all deadlines and it is easy to switch between monthly, weekly, and daily views.
- ❑ Implement a notifications section for all classes enrolled.
  - ❑

- ❑ Implement a user icon with picture that redirects to settings page.
  - ❑ The user is able to upload an image that will replace the default icon. Clicking it redirects them to the settings page.
- ❑ All features should be responsive on all devices.
  - ❑ The features look good on all devices, and the sidebar automatically closes on smaller screen sizes. (Calendar problem talked about in problem section below)

## User Story #7:

### **As a new teacher, I would like to have landing page help me set up a classroom.**

- ❑ Use our side bar with teacher's current classes.
  - ❑ The sidebar shows all of the teacher's current classes in an easy-to-view way.
- ❑ Implement the main section of the page to have directions on how the set up will work.
  - ❑ (See Below Section)
- ❑ Take user to a setup page where they can add the class information.
  - ❑ The setup page is easy to follow and contains all pertinent information.
- ❑ Teacher can then pick which sections her classroom would like (In-Class Lessons, Homeworks, Quizzes, Tests, Discussion board, Announcements).
  - ❑ The teacher is presented with checkboxes that allow them to include whichever features they would like to have in their class.
- ❑ Teacher should then add a first announcement message.
  - ❑ Instead of forcing teachers to add the message, we decided to allow them to add a message whenever they want to. (See below section)
- ❑ Add a picture to the classroom description.
  - ❑ The functionality to add a classroom picture is there, but there is no where to display it yet. (See below section)
- ❑ Auto-generate a class code that students can enter to join the classroom.
  - ❑ A six-digit class code is generated and testing to ensure that it is unique.
- ❑ Test to ensure classroom information is handled properly.
  - ❑ All classroom information is stored properly in Cloud Firestore and is formatted in a way that is easy to access.

## User Story #8:

### **As a student, I would like to be able to join my current classes.**

- ❑ Implement an add classroom section in the user settings page.
  - ❑ The student is able to enter the class code in the classroom settings page.
- ❑ Allow user to input code given by the teacher and click join.
  - ❑ The input box for the class code is easily visible.
- ❑ Test to make sure the code is valid.
  - ❑ When they enter a code, an error message is displayed if no code was given, the code is invalid, or the student is already enrolled in the class.
- ❑ Add classroom joined to the student's list of classrooms and sidebar.
  - ❑ Upon entering a valid code, the new class will show up in the class list and on the sidebar.
- ❑ Test to make sure the class is joined and updated properly.
  - ❑ After joining a class, the database is updated instantly and correctly.

## User Story #9:

**As a user, I would like to have an about page when first visiting the site.**

- ❑ Create a responsive navigation section.
  - ❑ The about page is able to resize based on the screen size and also has a nav bar which resizes too.
- ❑ Implement a section where the user can select which role they play in a classroom and take to specific section.
  - ❑ The About page includes a navigation bar which allows users pick which role they play in a classroom and then takes them to that section.
- ❑ Have a student information section detailing how Scribe Scholars can make a student's life easier.
  - ❑ The student information section explains why Scribe Scholars is useful for a student.
- ❑ Have a teacher information section detailing how Scribe Scholars can make a teacher's life easier.
  - ❑ The teacher information section explains why Scribe Scholars makes a teacher's job much easier.
- ❑ Have an administrator information section detailing how Scribe Scholars can make an administrator's life easier.
  - ❑ The administrator information section explains why Scribe Scholars makes a administrator's job much easier.
- ❑ Have a parent information section detailing how Scribe Scholars can make a parent's life easier.
  - ❑ The parent information section explains why Scribe Scholars makes a parent role much easier.
- ❑ Implement a contact us section for people to ask any questions they have.
  - ❑ The bottom of the page displays our team email and encourages people to reach out to us.

## What did not go well?

### User Story 3:

**As a user, I would like to be able to reset my password.**

- ❑ Set up Firebase to send password reset emails
  - ❑ We have the reset password working in the settings page but it should also be in the log in page as well.

### User Story #6:

**As a student, I would like to be able to have an easy to navigate landing page**

- ❑ All features should be responsive on all devices..
  - ❑ When the user has a small screen the calendar did not look very good so I choose to remove it. After being graded we decided it would be good if we had just a list of the dates on smaller screen sizes.

### User Story #7:

**As a new teacher, I would like to have landing page help me set up a classroom.**

- ❑ Implement the main section of the page to have directions on how the set up will work.
  - ❑ This task did not get completed because we had yet implemented the teacher landing page. For the sake of sprint one we decided for the student and the teacher to have the same landing page but in future sprints we plan to have the teacher have a personalized page. When they are new, before loading the class data they will be prompted to create a class.
- ❑ Teacher should then add a first announcement message.
  - ❑ We implemented the functionality to add an announcements but decided against making them create an announcement when the class is first created. They rather can just add them when they please.
- ❑ Add a picture to the classroom description.
  - ❑ The functionality to add a classroom picture is there, but there is no where to display it yet.
- ❑ Test to ensure classroom information is handled properly.
  - ❑ Currently when bad input is submitted it prints to the console but we still need to implement an alert to let the user know what went wrong.

### User Story #9:

**As a user, I would like to have an about page when first visiting the site.**

- ❑ Implement a contact us section for people to ask any questions they have.
  - ❑ We initially wrote that we were going to put a form on the button for users to contact us but we changed our mind and decided to put just our team email instead.

# How should you improve?

## General:

### **Not enough automated testing**

- ❑ Being that most of our team was new to the technologies we were using we spent most of the time developing and learning and did not learn how to implement testing into it quite yet. We do plan on starting that in the next sprint. As far as stuff that does not need unit testing we will continue to update our manual testing document with tests we run by hand such as a lot of the UI responsiveness testing.

### **Overestimating hours for a lot of tasks**

- ❑ While some of this was done on purpose for the point of us learning the technologies, we will be sure to shorten the number of hours for some of the UI tasks.

### **More user stories and shorter length user stories**

- ❑ My team and I did not know we were supposed to have 2.5 user stories per person, we had just thought the hours had to add up to 180. So for the next sprint we will be sure to have both of those requirements met without overestimating the hours or user stories.

### **Have a consistent alert messages for when user inputs bad data.**

- ❑ Since our team was using a couple different frameworks when working some of this sprint, some frameworks automatically handled error messages in their own way. We will be sure to make sure they are all consistent that way the UI is not jumping from error to error message format.

### **Not procrastinating until last few days of the week**

- ❑ With all of our team being in CS 252 and CS 307 it was a struggle trying to get all the Shell requirements and this project done with all the same deadlines. With the next coming weeks we are going to try to pace out the work a bit more so we don't have to do it all on the weekend.