Schedule with Descriptions
Team Name: Sizzling Syntax

FEBRUARY 8TH

Basic HTML Website Layout - Wiring a basic website design. All in HTML and as simple as possible. Layout should include navigation buttons and screens associated with the page.

UNIT TESTING

1. Test navigation between pages to make sure they load correctly and are accessible.

FEBRUARY 15TH

Plan DB Primary/Foreign Keys - Determining which keys (primary and foreign) that we will use for our database.
Create DB Schema - Setting up the database schema.
Website Navigation - Determining how users will navigate our website. Will not go into too much detail, but will contain how the user will be able to navigate the website and use the various functions that it will be able to do.
Set Up Domain for Website - Setting up domain for server.
Setting up Apache - Setting up the server and default runtime.

UNIT TESTING

1. Test the domain for the website.
2. Domain of website is accessible.
3. Navigation on AWS site is working correctly.
4. Simple setup of Apache.

FEBRUARY 22ND

Plan DB Control and Logic - Setting up the EC2 virtual instance on Amazon Web Services. Having the security group setup on the Amazon Web Services so we can shell into the virtual machine. Setting up the code deployment on Amazon WEb Services so that it recognizes our repository on GitHub. Create the secret and access keys for code deployment and ssh. Having the functionality of the login page completed.

Unit Testing:

1. Ensuring that the login page handles correct and incorrect usernames and passwords. Takes the user to the home page if fails, and displays the username.

Demonstration:

1. Show the EC2 Instance of the virtual machine on amazon
2. Show that the website can be accessed by www. And not including the www.
3. Have Apache server on the virtual machine to access the website
4. The login page functionality and handling if the username and password doesn't exist

---

MARCH 1ST

DynamoDB Setup - Setting the DynamoDB up according to the schema to be ready for testing and integration with Node.JS. Initial testing of receiving HTML requests will be done.
Start development Node.JS App - Begin developing the core code of our website. The Node.JS App will be what our game is run off of.
Integration of Node.js on Port 3000 - Integrate node.js web app (app.js) to port 3000 on aws
Signup Form: Signup Form for creating a new account and putting info in DYNAMODB.

UNIT TESTING

1. Test the database with the web application to see if the front end and back end work together.
2. Have our Node.JS application send and receive requests from the browser.
3. Allow DynamoDB to access our web application and handle HTML requests.
4. Validation of Sign up Form

Use Case testing for Signup Form:

1. Checks for the password length of six characters
2. Checks to see that at least one capital in password.
3. Username of length six (using regular expressions)
4. Sign up button initiates the validation and makes sure all fields are not blank

Demonstration:

1. Show that the website now listens on port 3000
2. Our new sign up form page and aesthetics of the page
3. Our redesigned top bar and new backgrounds/buttons on the HomePage.
4. Showcasing the signup form and its validation
5. The submit button populates the DynamoDB ‘user’ table in our database.
6. The form then redirects back to the Home page

MARCH 8TH

Isolate: Installing Isolate sandbox environment on github, and successfully deploying it on the Amazon Web Services
Isolate Boxes: Creates a box in order for the user to store a symlink for their language
Isolate Code: User can select the language that they wish to solve challenges in
Implement Ace Code Editor Web IDE - Create and implement the online IDE environment. Complete methods for verifying output.

UNIT TESTING

1. Test the IDE and make sure it sends the correct file to sandbox.
2. See if compilation in the IDE is properly functioning

MARCH 15TH

Node.JS and Isolate Integration - Get Isolate and the Node.JS App working together properly.
DynamoDB AND Node.JS Integration - Will integrate DynamoDB and Node.JS into Isolate to be able to run what we create within the sandbox with no compatibility issues.
Testing Ace Code Web IDE - Ensuring the DBMS is functioning properly based off the user's source code. Make sure there are no security vulnerabilities. Also making sure all the acceptable languages are working properly.

UNIT TESTING

1. Test the IDE with the website and embed it as well as run more tests to sandbox.
2. Running and executing code in our web application on Isolate
3. Console logs whether or not the get request from the server side has the correct output
4. The server side takes the selected language, creates the proper symlinks, and then runs the code accordingly.

Demonstration for March 8th - 15th

1. Added footer to the home page
2. Ace Code Editor embedded into our Practice Mode Page
3. Practice Mode Page can select your language
4. A user can select from the dropdown menu which language they wish to use
5. Symlinks are used to pass the javac and java calls for the program to be compiled and executed.
6. The console logs whether or not the code submitted was correct or incorrect

---

MARCH 22ND

Website Front End Finalization - Finalizing what our website will look like. Very detailed step, and will be what the website will look like to the users.
AsyncTasks- Fixed functions in node because asynchronous the way it was set up would have created problems down the line.

UNIT TESTING

1. Final touchups of front end and database testing for the website.
2. Contact Us page form for emailing
3. Login in functionality for local host

Demonstration/Code for March 22nd

7. Added footer to the home page. New buttons on the home page
8. Overhaul of async callbacks and wrote new functions for the isolate sandbox environment
9. Checking if username already exist in the database

MARCH 29TH

Setup Master Account List - Setup database to contain a table that handles all user accounts.

Implementation of DBMS Functions - implement database functions to connect and control database.

UNIT TESTING

1. Setup the master account list for database and finish up database management
2. DBMS can access the Master Account List and vice versa

APRIL 5TH

DBMS, Web App, and Isolate Compatibility - interface the database web app and sandbox to all work together

Create Admin Accounts for Development - Admins will create accounts to be able to test the core functions of the game.

UNIT TESTING

1. Setup admin account and make final tests for compatibility with website and sandbox.

APRIL 12TH

Create List of Problems - Create a custom list of problems for the user to solve in our game.

Create Problem Descriptions - Create a description for the problems that we create so that we and the user will be able to better understand what he/she is supposed to do.

UNIT TESTING

1. Test logic of problems to ensure that they are appropriately challenging.

APRIL 19TH

Sandbox Stress Test - Testing the sandbox against test cases of programs.

UNIT TESTING

1. Using our list of problems, attempt to break the sandbox from functioning properly.

APRIL 26TH

Testing Functions for Groups - Ensuring all functions within the group environment can be called properly.

Testing Functions for Admins of Groups - Ensuring all admin functions (group settings, create/delete group, etc) can be called properly.

Testing Functions for User Accounts - testing functions that control user account settings and functionality

UNIT TESTING

1. Try creating a group, tamper with settings, and delete it.
2. Try creating an account, tamper with settings, and delete it.
3. Test user interactions with an existing group.

MAY 3RD

Promotional Website - Website for our demo showcase.

Final Website Deployment - Our final version of the website. Will be everything that the user will be able to interact with and play our game.

UNIT TESTING

1. Navigation on both websites to ensure user is able to access all pages.
2. Test all user interaction features.
3. Test multiplayer connectivity.