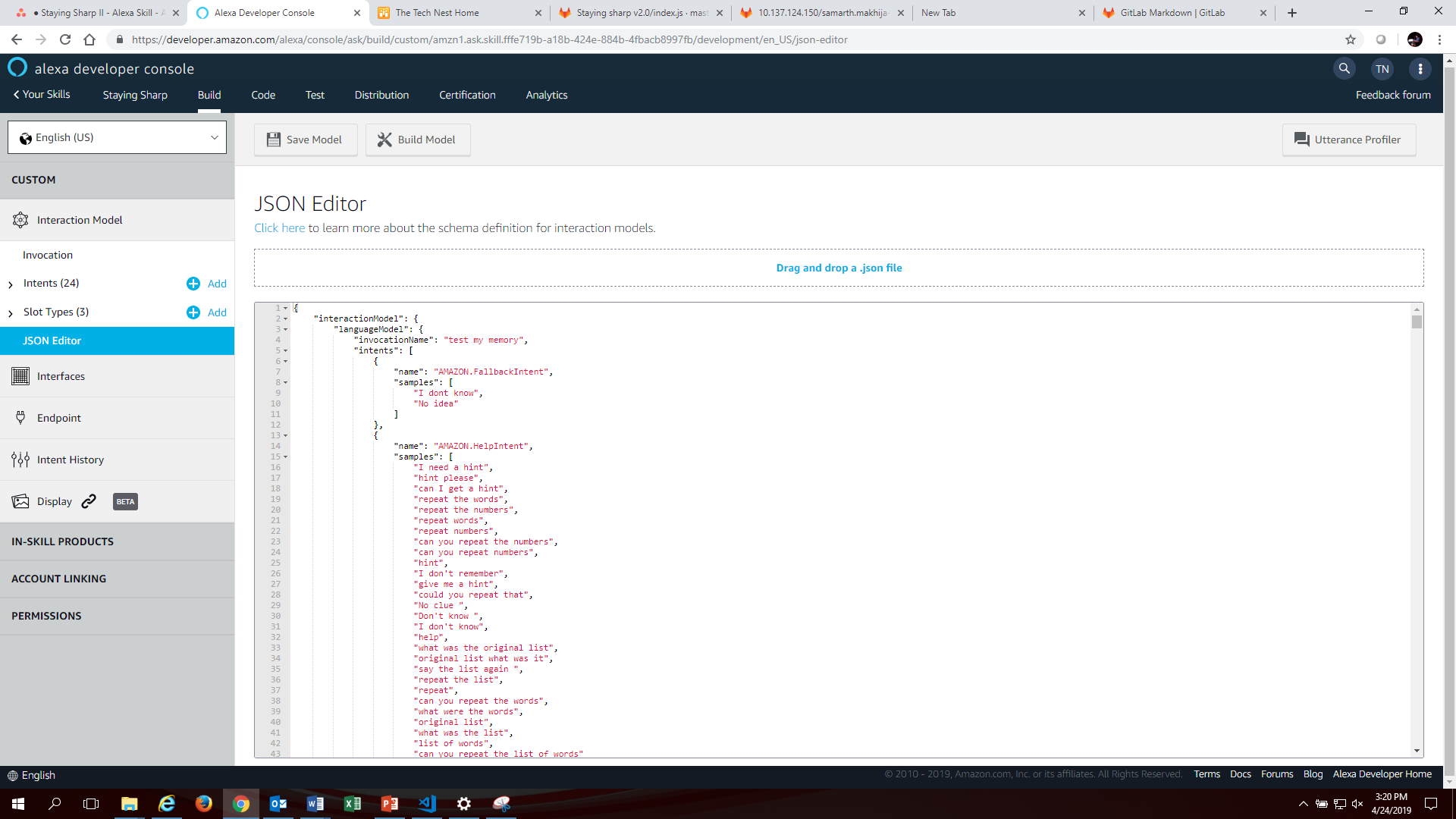
**Documentation**

1. **Deployment Package**
   1. The deployment package is a compressed folder that contains all the files relevant to the project. The .zip can be uploaded directly to AWS Lambda in order to get the project’s back-end up and running. The front\_end.json file on the other hand, can be copied into the Alexa Developer Console in order to get access to the interaction models for the skill.
   2. Explanation of files
      1. ***Index.js***
         * This is the main file for the program that handles all functionality, features and actions for different intents.
      2. ***Front\_end.json***
         * This file is the front-end schema that gets put onto the Alexa Developer console and makes for the voice recognition and natural language processing aspect of the program.
      3. ***Package.json***
         * This is default file generated by AWS Lambda that ensures that the right Alexa SDK is imported
      4. ***Dictionary.js***
         * Contains all the different words and categories that form the database for the “words” game. This also contains code that takes input from the main program and returns the list of words to remember and the list of words that are used to question the user, with just one right answer.
      5. ***ChunkingDictionary.js***
         * Generates the random list of numbers based on the difficulty level set by the program
   3. Index.js Code documentation
      1. **Variables:**
         * **Dialogues:** WELCOME\_MESSAGE, HELP\_MESSAGE, HELP\_REPROMPT, STOP\_MESSAGE, GAME\_SELECTION\_MESSAGE, INSTRUCTIONS\_MESSAGE, CONFUSED\_REPROMPT, wordTIPS, numTIPS, HELP\_MESSAGE\_NUMBERS, compliments, motivations
         * **Instructions:** Boolean to check if the program is coming from the instructions intent
         * **previousTipIndex:** The index of the previous tip in order to keep them dynamic
         * **transitionMessage:** Variable to store the fragments of the message that Alexa has yet to speak in order to maintain dialogue consistency
         * **lastQuestionMessage:** Similar to transitionMessage
         * **DEFAULT\_WORDS\_TO\_REMEMBER\_LENGTH:** Same as name
         * **DEFAULT\_WORDS\_TO\_PICK\_FROM\_LENGTH:** Same as name
         * **MAX\_NUMBER\_TURNS:** Maximum number of rounds that can be played in the words game
         * **MAX\_NUMBER\_TURNS\_CHUNKING:** Maximum number of rounds that can be played in the numbers game
         * **numCorrectAnswers:** Tracks number of correct answers overall
         * **num\_CorrectWordAnswers:** Tracks number of correct word answers
         * **num\_CorrectNumAnswers:** Tracks number of correct chunking exercise answers
         * **numIncorrectAnswers:** Tracks number of incorrect answers overall
         * **num\_IncorrectWordAnswers:** Tracks number of incorrect word answers
         * **num\_IncorrectNumAnswers:** Tracks number of incorrect chunking exercise answers
         * **gotchunkingQuestionCorrect:** Flag to check if the user got the chunking exercise question right
         * **isChunkingGame:** Flag to determine which type of game the user is in
         * **isWordGame:** Flag to determine which type of game the user is in
         * **leftoverChunkingRoundSpeech:** Variable to handle leftover fragments to maintain dialogue consistency
         * **percent\_NumCorrect:** Chunking game score
         * **percent\_WordCorrect:** Word game score
         * **cardObject:** Logos for visuals
         * **tell\_user\_speech:** Used to restart the game by adding sorry I didn’t catch that when user says something unexpected at initiation
         * **currentDifficulty:** Determines the current level of difficulty
         * **level:** Used to calculate difficulty
         * **round:** The number of rounds played within current game
         * **numTurns:** The number of total turns within a game
         * **chunkingWordLength:** Default number of words to remember
         * **wordsList:** List of three words that has one answer
         * **wordsToRemember:** List of (7/10/12) words that make up the answers to the questions
         * **exerciseSelectionFlag:** Determines whether an exercise has been selected or not
         * **randomNumber:** Same as Name
         * **isTip:** Whether the previous intent was tip intent or not
         * **finished(Words/Numbers)Game:** Status of games
         * **endgame:** True if both games have been played and game needs to end
         * **fallback:** True if the previous intent was the fallback intent
         * **wordsToPickFrom:** Same as wordsList
         * **setofInstructions:** Depending on context contains the instructions to be said
         * **levelupspeech:** Maintains continuity on increasing difficulty
      2. **Function Descriptions are within the code itself**
2. **Front End**

The front end can be copied straight from the json file and then pasted onto the Alexa Developer Console.



1. **Explanation of Intents:**

* **Exercise Selection:** Used for determining the game type
* **Start:** Used to start the game/invoke the skill
* **UserNumberAnswer:** Takes in the numbers to determine whether the user answered correctly or not
* **UserAnswer:** Takes in the words to determine whether the user answered correctly or not
* **Tip:** Gives users tips around the game to improve their performance
* **Restart:** Used for restarting the game
* **Instructions:** Gives users context on the game as well as all the commands they can Alexa to control different aspects of the game
* **Amazon.Fallback:** Handles the unhandled inputs when user says something the game isn’t programmed for
* **Amazon.Help:** Reminds users the original list in case they forget it
* **Amazon.Stop:** Stops the program no matter the stage and tells users their score
* **Amazon.NavigateHome:** Same as restart
* **Amazon.Yes/No:** Handle yes and no dynamically based on the previous intent the game was in by making use of flags