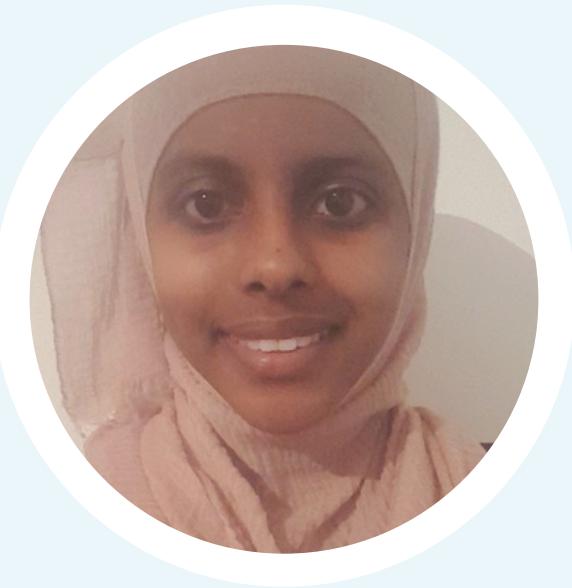


The Imitation Game



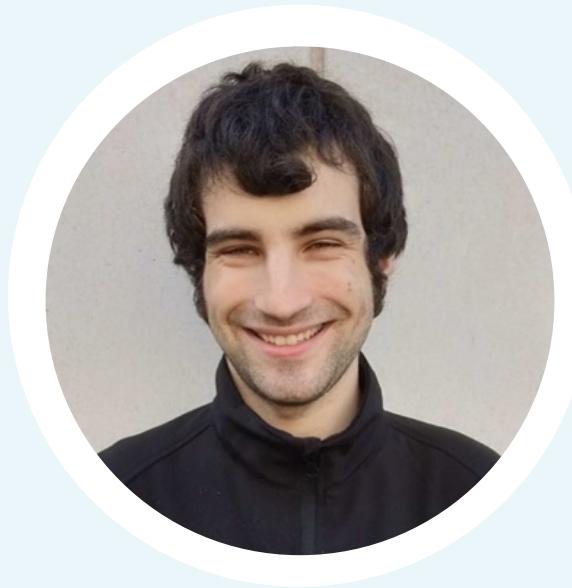
The Team



Suad



Rachel



Alex



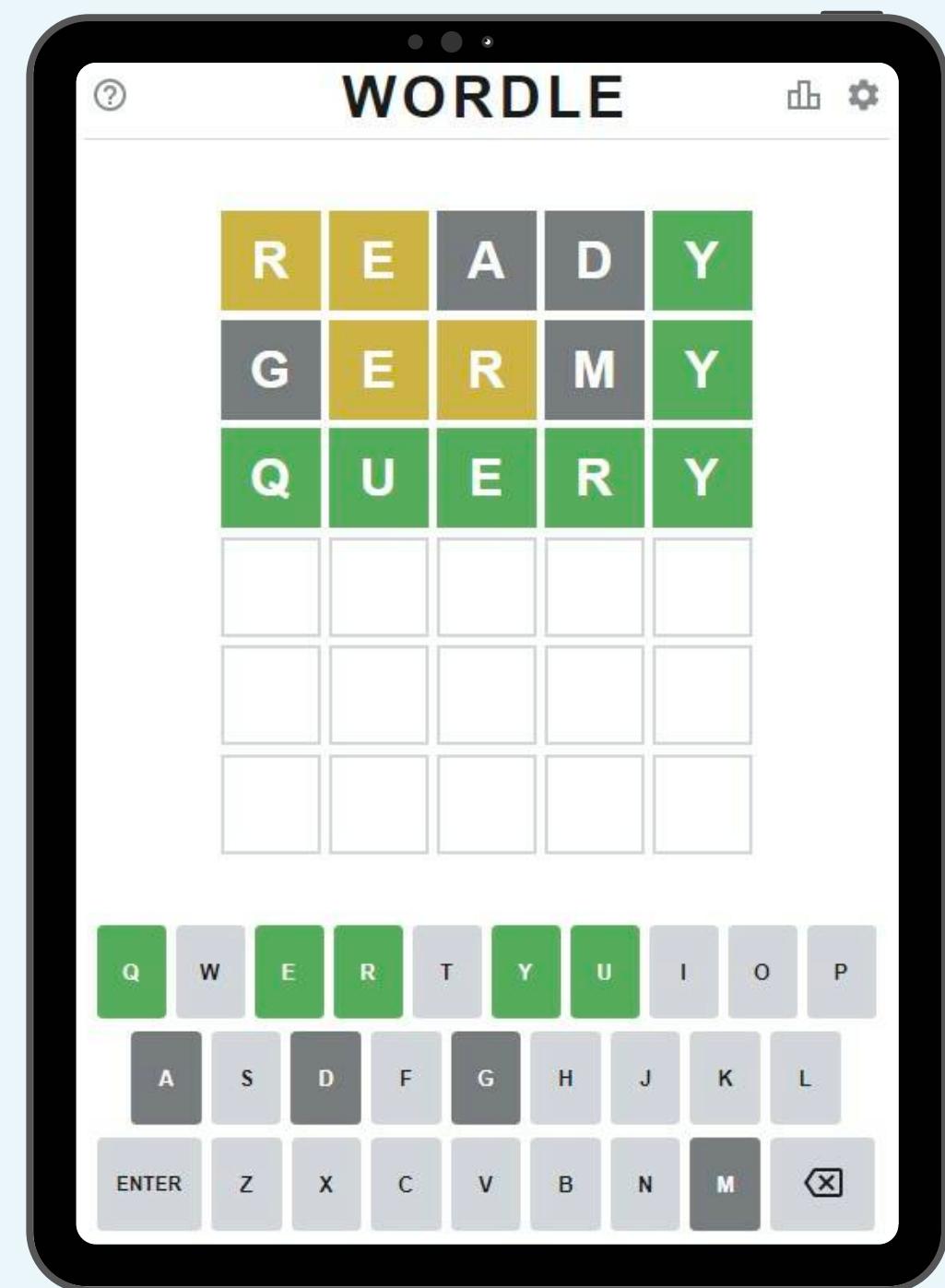
Cristian



Rosalinda

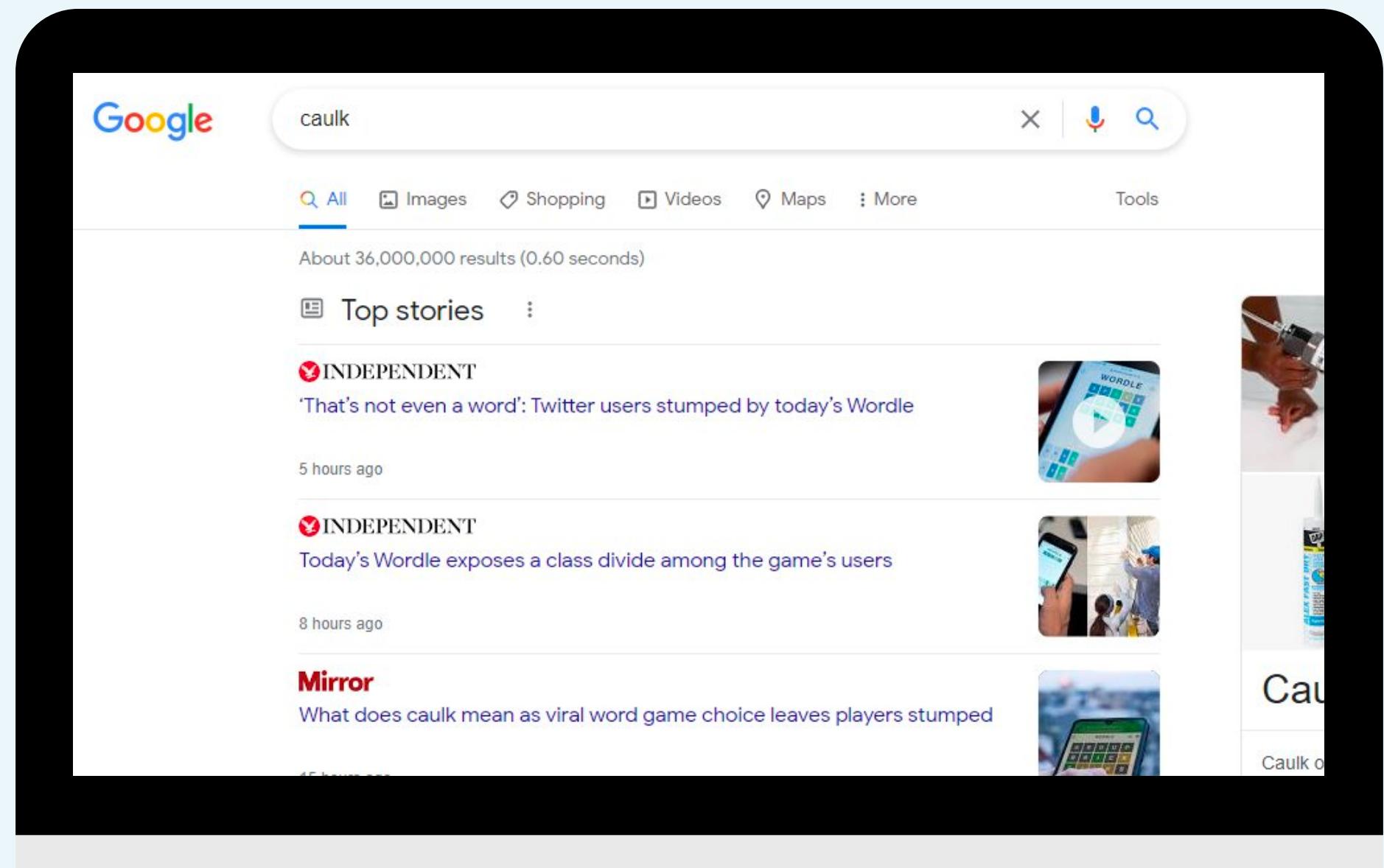
W O R D L E
H E L P E R

What is Wordle?



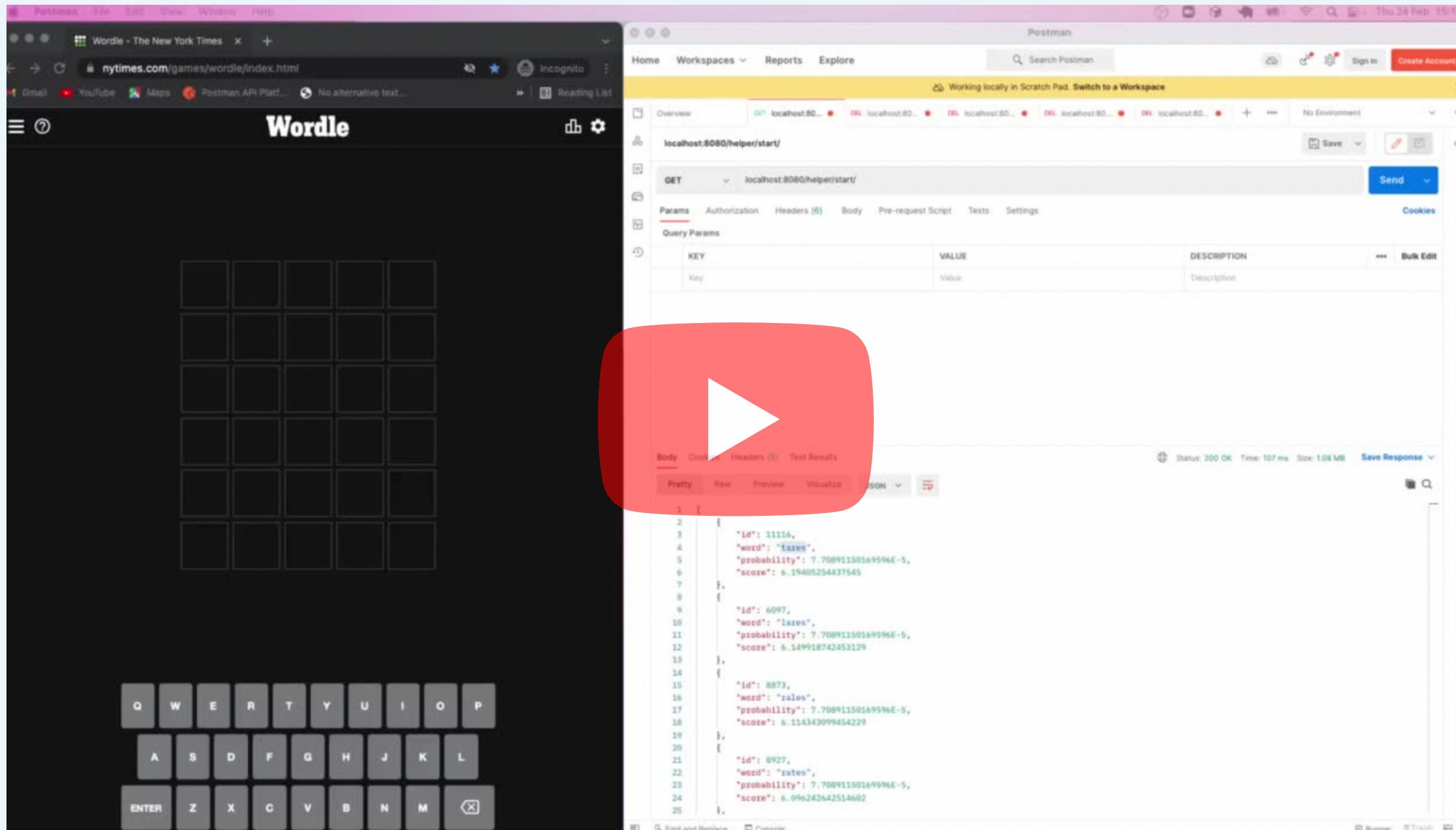
W O R D L E
H E L P E R

The Problem



Helper Mode

(Click below for video link)



Competitive Mode

User VS Machine

GET <localhost:8080/competitive/userresults/ZoomHost1234>

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "actual_word": "other",
4     "userName": "ZoomHost1234",
5     "guessesTaken": 5,
6     "machineGuesses": 3
7   },
8   {
9     "actual_word": "thorn",
10    "userName": "ZoomHost1234",
11    "guessesTaken": 6,
12    "machineGuesses": 5
13  },
14 ]
```

User Average Score

GET <localhost:8080/competitive/averageresults/BigBird>

1

Body Cookies Headers (5) Test Results

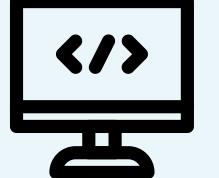
Pretty Raw Preview Visualize JSON

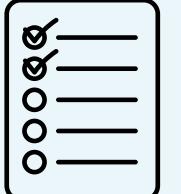
```
1 2.6
```



Our Approach

 Highly collaborative

 Mobbing and pair-programming

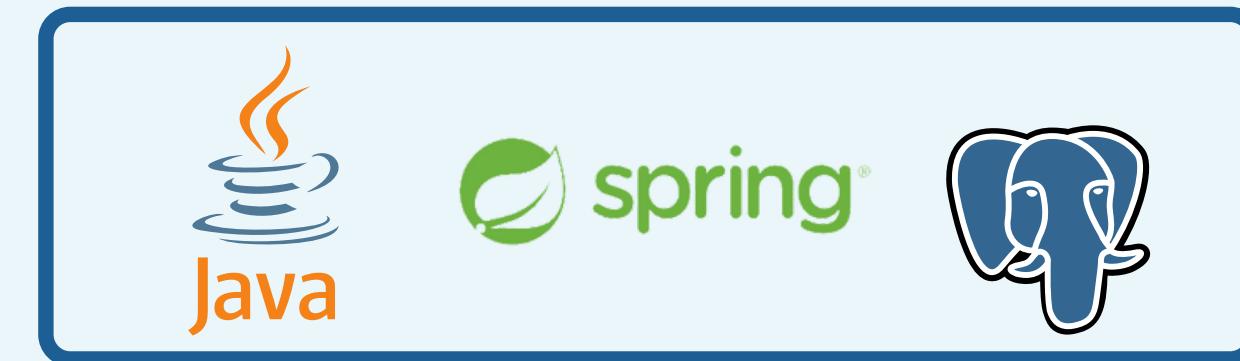
 Planning as a group

Planning

Date	Tasks
Sunday (20/02)	<p><i>Whiteboard group planning against Requirements:</i></p> <p>POJOs, Controller Layer, Service Layer, DAO Layer, Tables</p>
Monday (21/02)	<p>Create ER Diagram</p> <p><i>Version 1:</i></p> <ul style="list-style-type: none"> • (Re)build version 1 of wordle solver (mob programming) • Write tests for methods in version 1 (pair programming)
Tuesday (22/02)	<p><i>Version 2:</i></p> <ul style="list-style-type: none"> • DAO Layer to access DB • Method to import from CSV to words table • Integrate with wordle database <p>Integration of an API (Controller), to send HTTP requests to progress game of Wordle</p> <p>Testing of all methods in Helper Mode</p>
Wednesday (23/02)	<p><i>Work on Competitive Mode:</i></p> <ul style="list-style-type: none"> • Have post request to allow users to create User Object (POJO), and add this to database • Create SQL tables for games played, users, and actual answers • Take the actual answers from the Wordle website, and match them to the date on the txt (csv) file • Re-use the function to import from txt (csv) file • Create new controller class for users • Create users package • Create user DAO and Data Access Service • Create user service • Add methods - e.g., playFullGame (returns number of guesses for the computer) • In DataAcessService, include method which uses Join • Add custom exception



 **diagrams.net**



Helper Mode

Original Word List	
PK	<u>originalword_id int SERIAL NOT NULL</u>
	word varchar(5) NOT NULL UNIQUE
	probability numeric
	score numeric

Users	
PK	<u>user_id int SERIAL NOT NULL</u>
	name varchar(50)
	email varchar(50) UNIQUE NOT NULL
	username varchar(50) UNIQUE NOT NULL

Competitive Mode

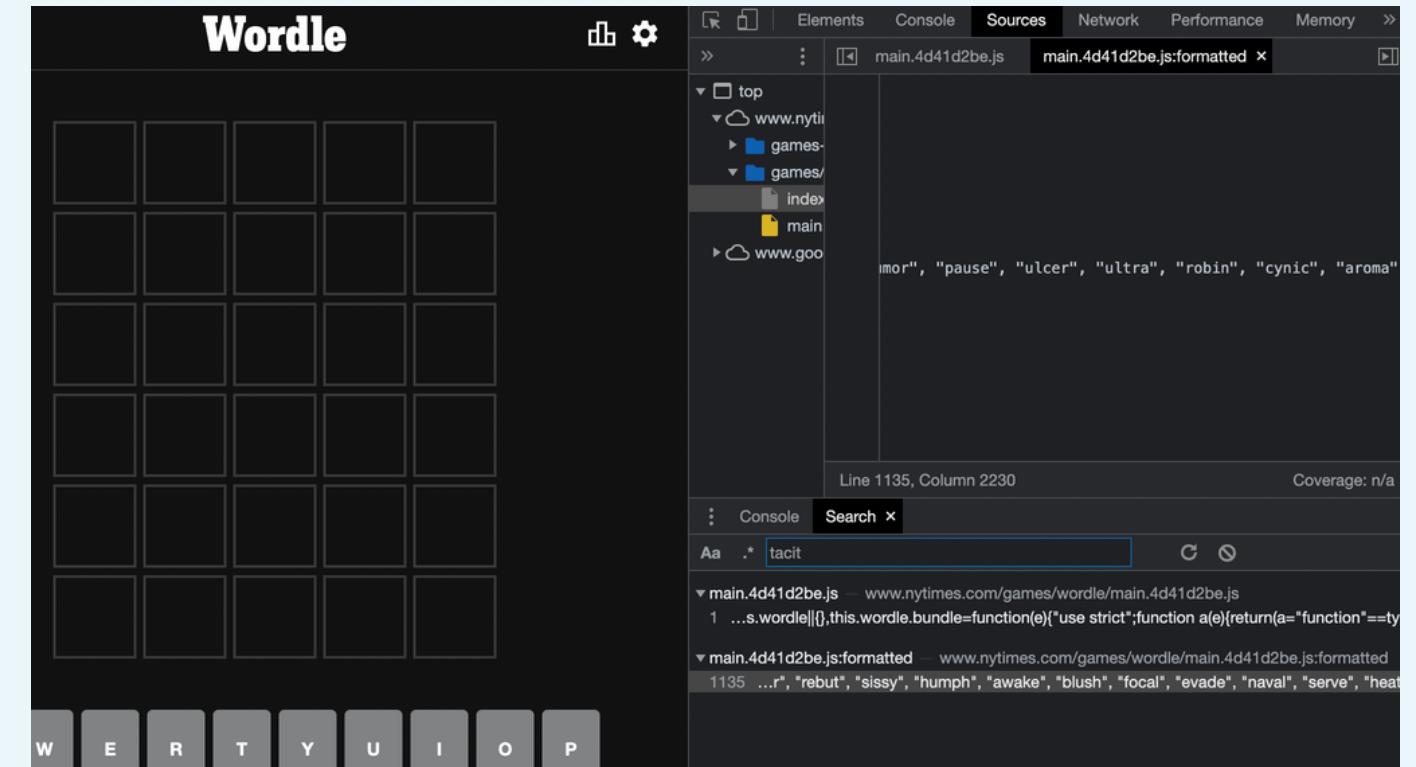
Games Played	
PK	<u>gamesplayed_id int SERIAL NOT NULL</u>
FK	user_id
FK	actual_answers id
	guesses_taken int

Actual Answers	
PK	<u>actualanswers_id int SERIAL NOT NULL</u>
	date_of_answer varchar(10)
	actual_words varchar(5)
	machine_result int

WORDLE
H E L P E R

Compute Machine Guesses

1) Obtain actual answers using inspect element



2) Populate SQL table with answers

3) Apply solver to each answer with a while loop to count number of guesses

```
Word guess;
while(running){
    numberofGuesses++;
    // Get highest scoring word from sorted guess list
    guess = guessList.get(0);
    // Generate pattern and find all matching words
    pattern = generateWordPattern(guess, wordAnswer);
    guessList = findMatchingWords(guess, guessList, pattern);
    // Recompute scores
    guessList = computeScoreDistribution(guessList);
    // Sort by score
    guessList = guessList.stream()
        .sorted(Comparator.comparing(Word::getScore).reversed())
        .collect(Collectors.toList());
    // Check if guess matches answer
    if (guessList.get(0).getWord().equals(answer.getAnswerOfDay())){
        numberofGuesses++;
        running = false;
    }
}
```



SQL

Original .txt file

```
1 aahed,7.70891150169596E-5,4.459449718932218
2 aalii,7.70891150169596E-5,3.9391774228771244
3 aargh,7.70891150169596E-5,4.1670388314296725
4 aarti,7.70891150169596E-5,4.800936023446364
5 abaca,7.70891150169596E-5,3.1522578815960625
6 abaci,7.70891150169596E-5,3.8945362838209086
7 aback,7.70891150169596E-5,3.5248854729761447
8 abacs,7.70891150169596E-5,4.351905582270924
9 abaft,7.70891150169596E-5,3.6601430976106903
10 abaka,7.70891150169596E-5,3.046985676568135
11 abamp,7.70891150169596E-5,3.582470460378561
12 aband,7.70891150169596E-5,4.010194463310032
13 abase,7.70891150169596E-5,4.696091046926338
14 abash,7.70891150169596E-5,4.072441992330873
15 abask,7.70891150169596E-5,3.9990941840831136
16 abate,7.70891150169596E-5,4.53848955721426
17 abaya,7.70891150169596E-5,3.104373456528726
18 abbas,7.70891150169596E-5,3.7458654302668535
```

Transferring to SQL

```
1 COPY original_word_list (word, probability, score) FROM
2 '/Users/Suad/Downloads/initialcalculations.txt' DELIMITER ',' CSV;
3
```

id	word	probability	score
1	aahed	0.0000770891150169596	4.459449718932218
2	aalii	0.0000770891150169596	3.9391774228771244
3	aargh	0.0000770891150169596	4.1670388314296725
4	aarti	0.0000770891150169596	4.800936023446364
5	abaca	0.0000770891150169596	3.1522578815960625
6	abaci	0.0000770891150169596	3.8945362838209086
7	aback	0.0000770891150169596	3.5248854729761447
8	abacs	0.0000770891150169596	4.351905582270924
9	abaft	0.0000770891150169596	3.6601430976106903
10	abaka	0.0000770891150169596	3.046985676568135
11	abamp	0.0000770891150169596	3.582470460378561
12	aband	0.0000770891150169596	4.010194463310032



SQL

Data Access Layer Code Snippet

```
private RowMapper<GameResults> gameResultsRowMapper = (rs, rowNum) -> {
    GameResults gameResults = new GameResults(
        rs.getString("actual_word"),
        rs.getString("username"),
        rs.getInt("guesses_taken"),
        rs.getInt("machine_guesses")
    );
    return gameResults;
}

public List<GameResults> getUserGuessVsMachineResultsListForDate (LocalDate date) {
    String sql = """
        SELECT actual_answers.actual_word, username, all_games.guesses_taken, actual_answers.machine_guesses
        FROM users
        INNER JOIN all_games
        ON users.id = all_games.user_id
        INNER JOIN actual_answers
        ON all_games.actual_answers_id = actual_answers.id
        WHERE actual_answers.date_of_given_answer = ?
        """;
    List<GameResults> gameResultsList = jdbcTemplate.query(sql, gameResultsRowMapper, Date.valueOf(date));
    return gameResultsList;
}
```

All Games SQL Table

actual_word	username	guesses_taken	machine_guesses
loojd	BigBird	5	2
loojd	SusuTheFLowerPot	1	2
loojd	kittyBeige	6	2



Testing

```
● ● ●

@Test
void canGetWordById() {
    // Given
    Word testWord1 = new Word(1, "hello", 0.25, 2.0);
    Word testWord2 = new Word(2, "image", 0.25, 1.0);
    Word testWord3 = new Word(3, "opens", 0.25, 0.5);
    Word testWord4 = new Word(4, "fares", 0.25, 0.25);
    List<Word> allWords = new ArrayList<>();
    allWords.add(testWord1);
    allWords.add(testWord2);
    allWords.add(testWord3);
    allWords.add(testWord4);
    given(wordDAO.selectWordById(1)).willReturn(testWord1);

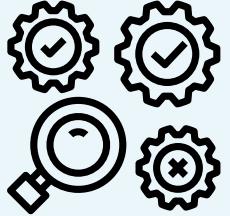
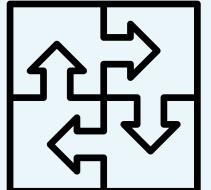
    // When
    Word actual = underTest.getWordById(1);

    // Then
    Word expected = testWord1;
    assertThat(actual).isEqualTo(expected);

}
```

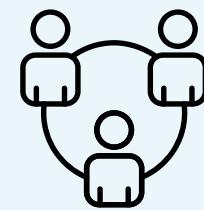
W	O	R	D	L	E
H	E	L	P	E	R

Future Developments

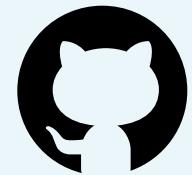
-  Include full user authentication
-  Testing all edge cases
-  Integrate with Wordle



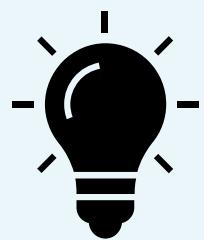
Key Learnings



Importance of planning and communication



Collaborating using Git



Ingrained all our training to date

Thank You !



(Github Link)

W O R D L E
H E L P E R

Extra Slides

W O R D L E
H E L P E R

Generating Word Pattern



```
70 public LinkedHashMap<String, String> generateWordPattern(Word word, Word targetWord) {
71
72     LinkedHashMap<String, String> pattern = new LinkedHashMap<>();
73
74     Character[] lettersInWord = new Character[word.getWord().length()];
75     Character[] lettersInTarget = new Character[targetWord.getWord().length()];
76
77     for (int i = 0; i < lettersInWord.length; i++) {
78
79         lettersInWord[i] = word.getWord().charAt(i);
80         lettersInTarget[i] = targetWord.getWord().charAt(i);
81
82         if (lettersInWord[i] == lettersInTarget[i]) {
83             pattern.put(String.valueOf(lettersInWord[i]) + i, "green");
84             lettersInWord[i] = null;
85             lettersInTarget[i] = null;
86         }
87     }
}
```

