

2019115021

Arvind P

Full Stack Project Report

Word Unscrambling Game using

Flask and MongoDB

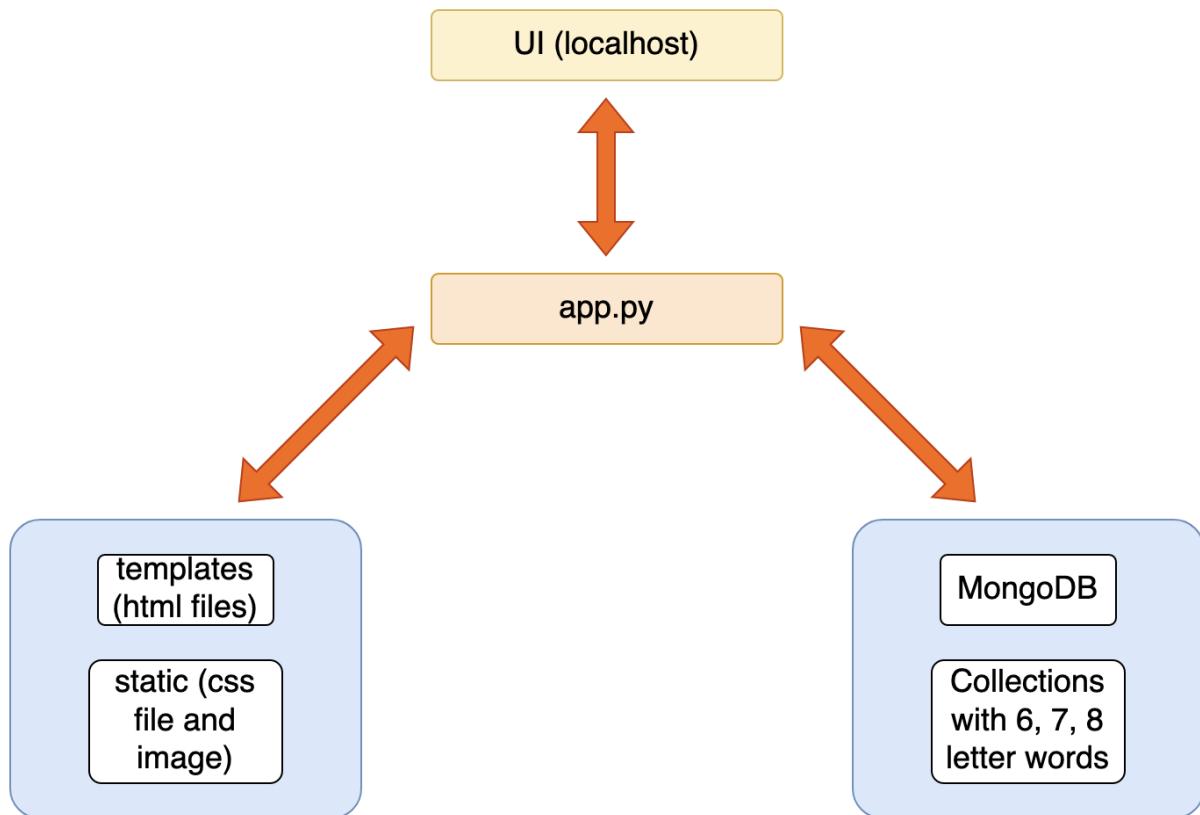
1. Abstract:

This is a web application that is a simple to play word unscrambling game. A scrambled word, representing a six, seven, or eight letter word, is presented to the player. The player has to find out the correct word as fast as they can. The application also comes with statistics which players can use to note their progress, and improve their best scores and times.

A python program made with Flask accesses HTML templates, a static CSS file, and a background image to render the UI along with Bootstrap. It also accesses words stored in three collections under a MongoDB database, scrambles them, and displays them for the player to unscramble. This application serves as an easy-to-play non-addictive exercise to help improve one's concentration and enhance their memory.

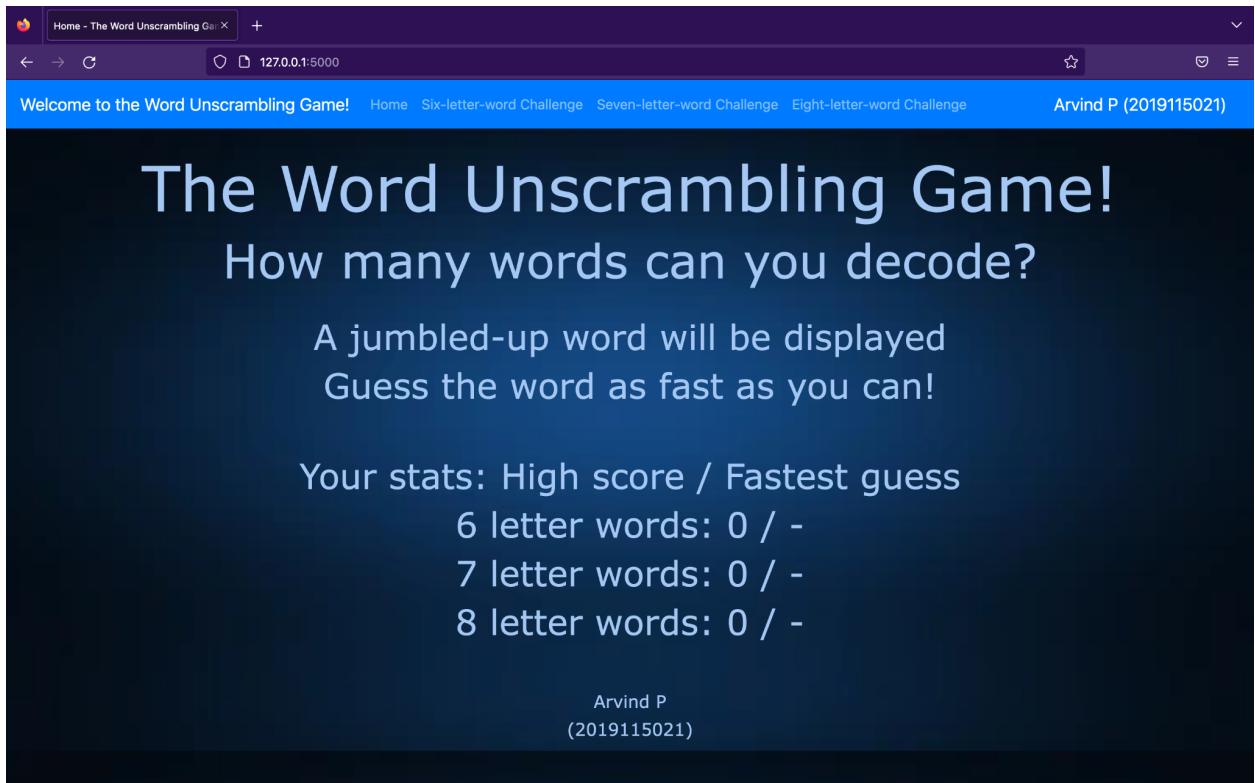
Keywords: Flask, MongoDB, HTML, CSS, Bootstrap

2. Architecture Diagram:

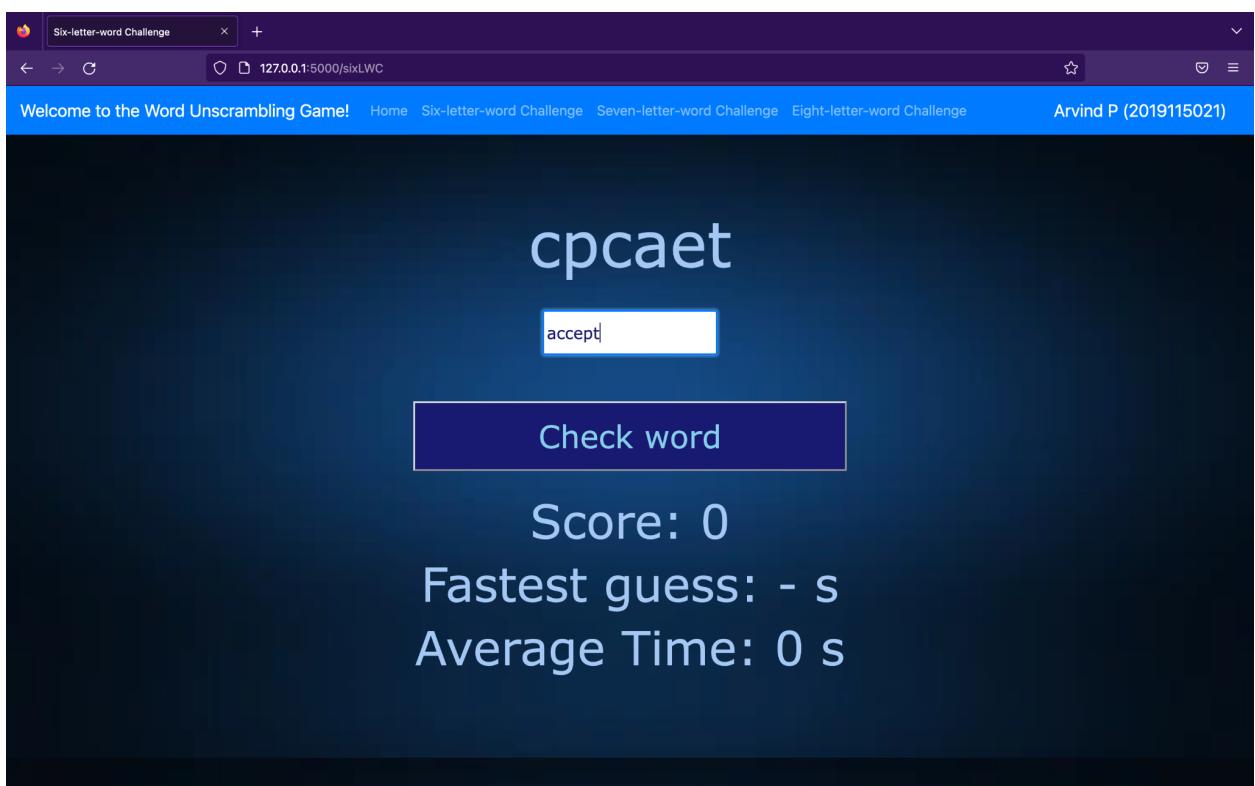
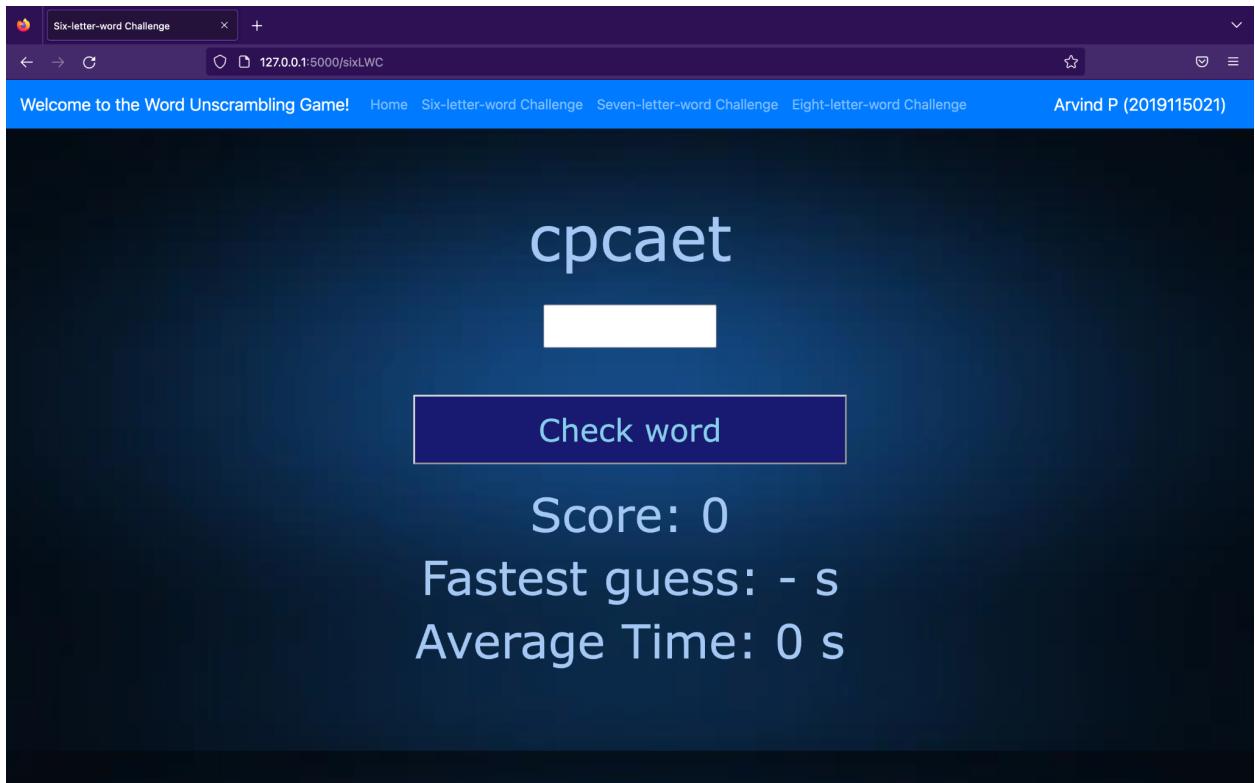


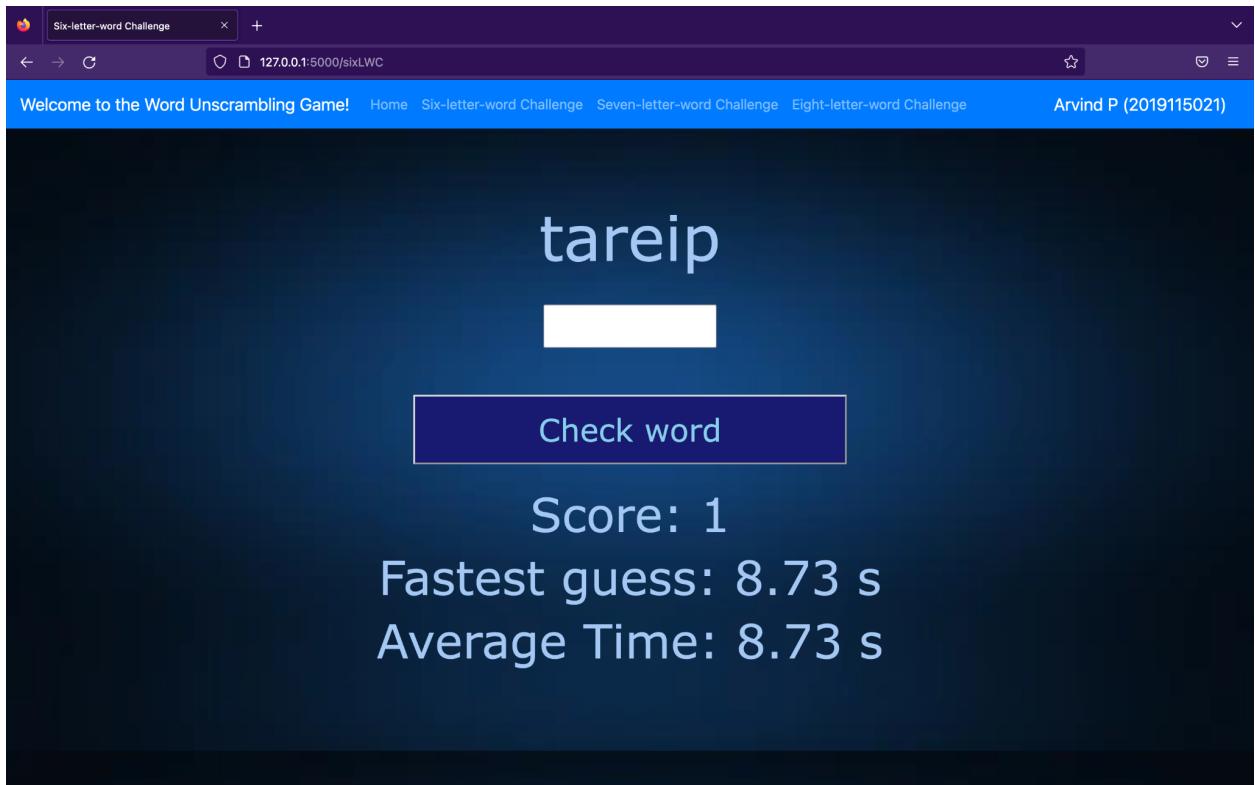
3. Screenshots:

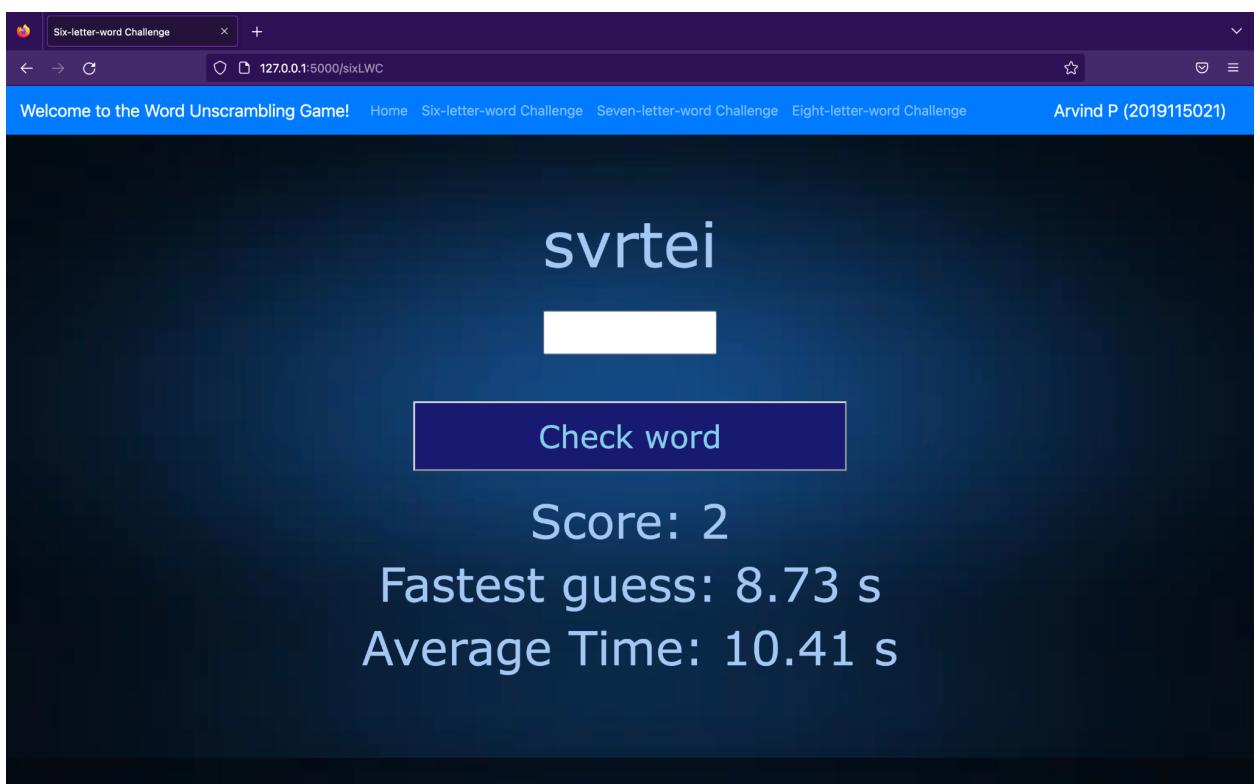
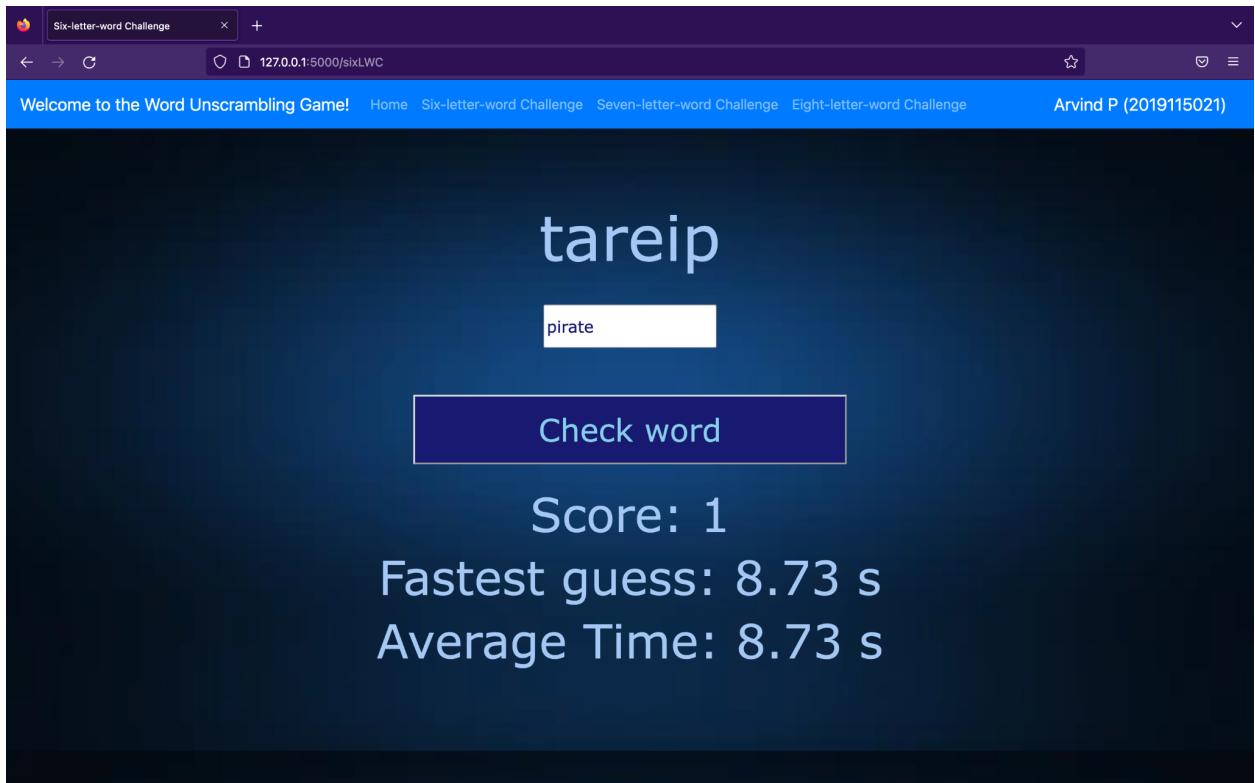
Home page:



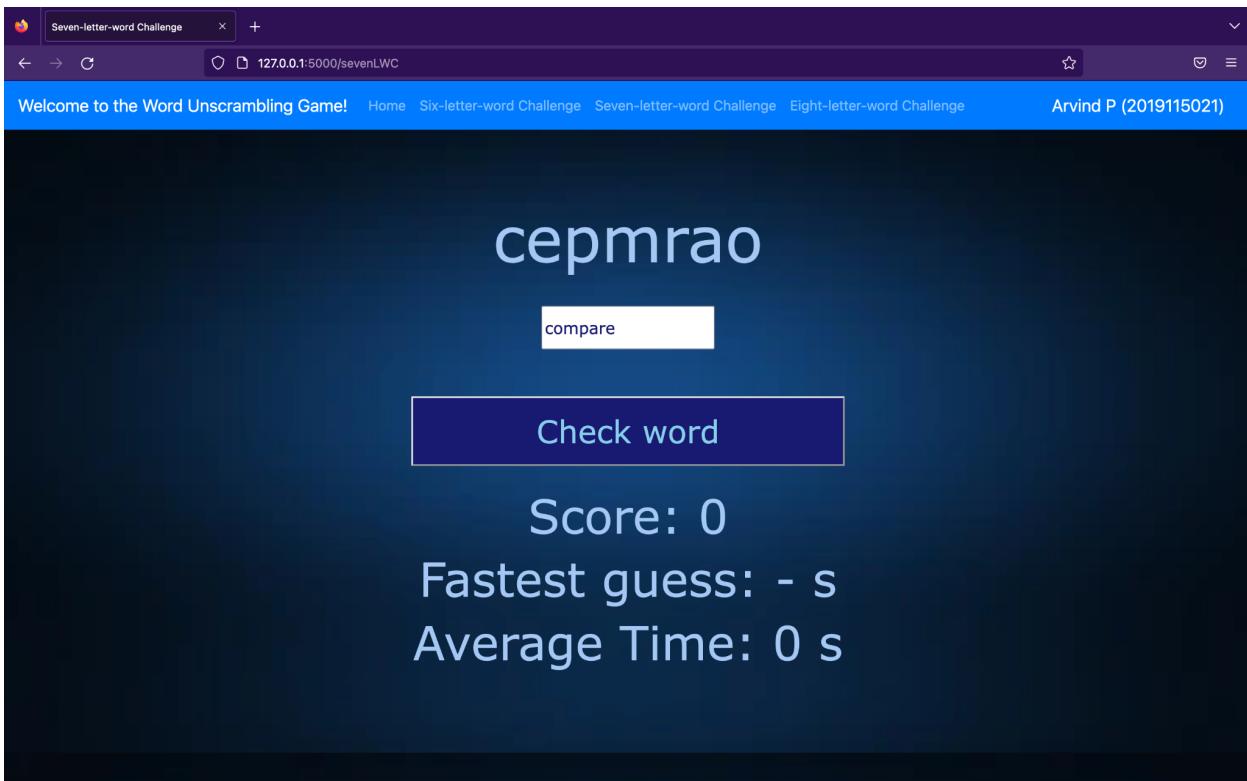
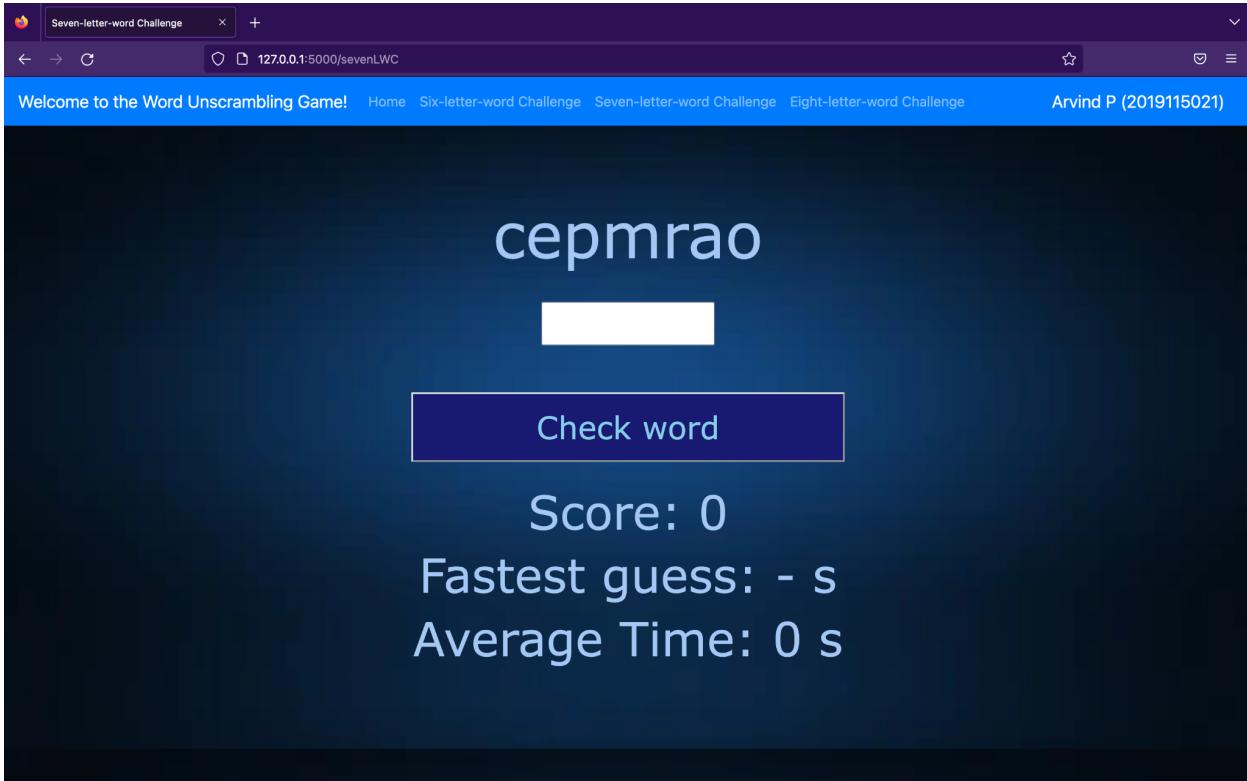
Six-letter-word challenge:

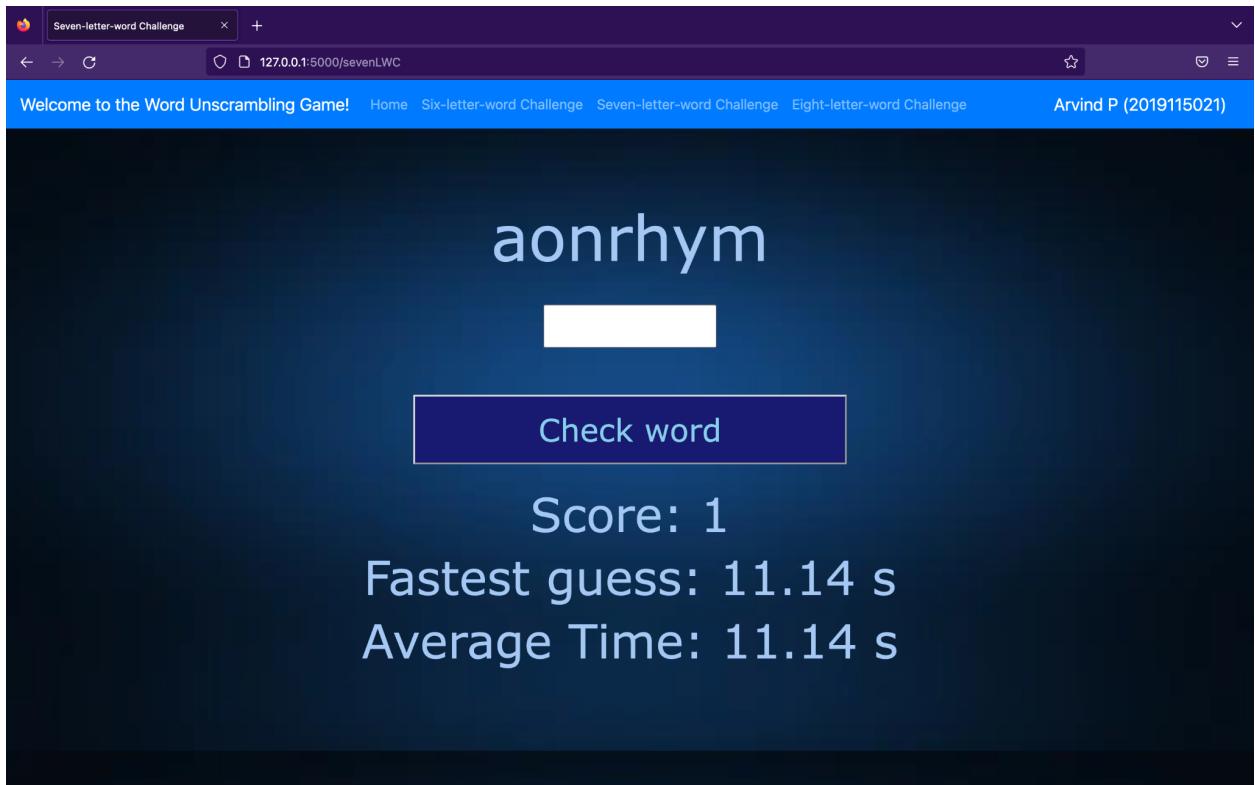


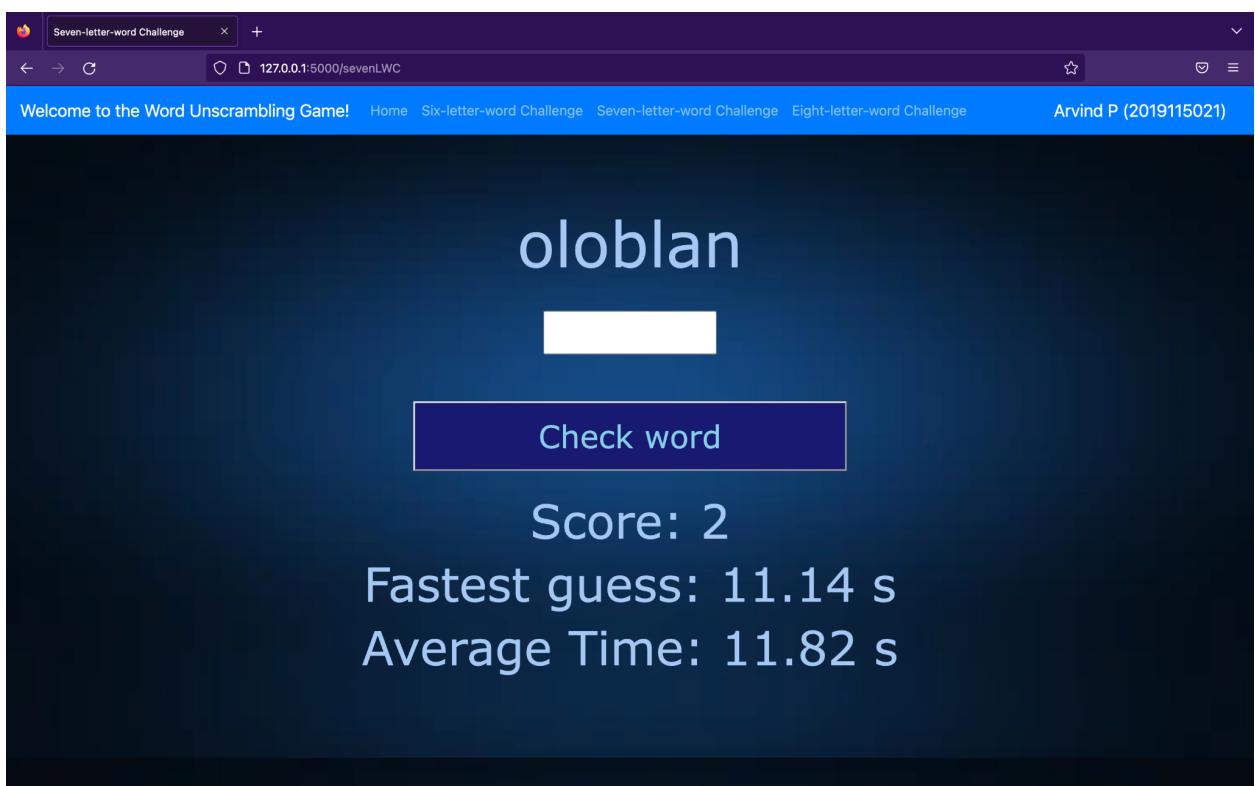
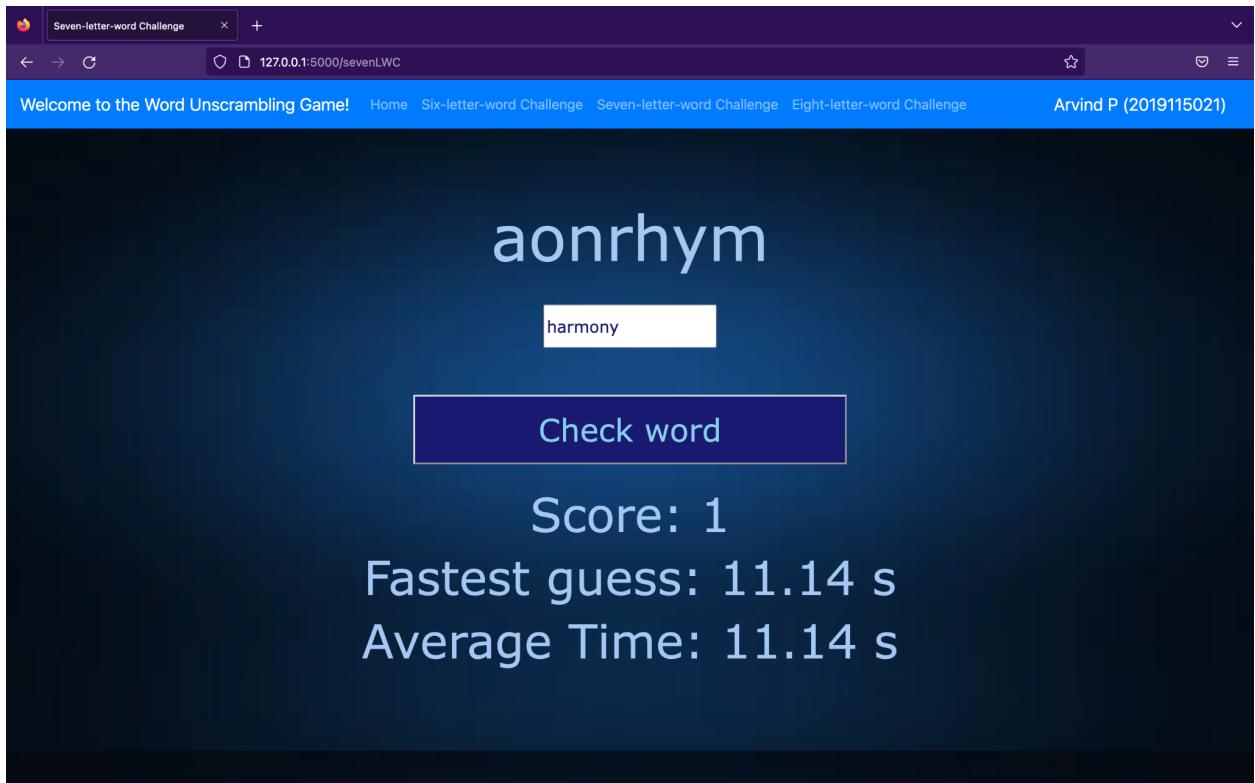




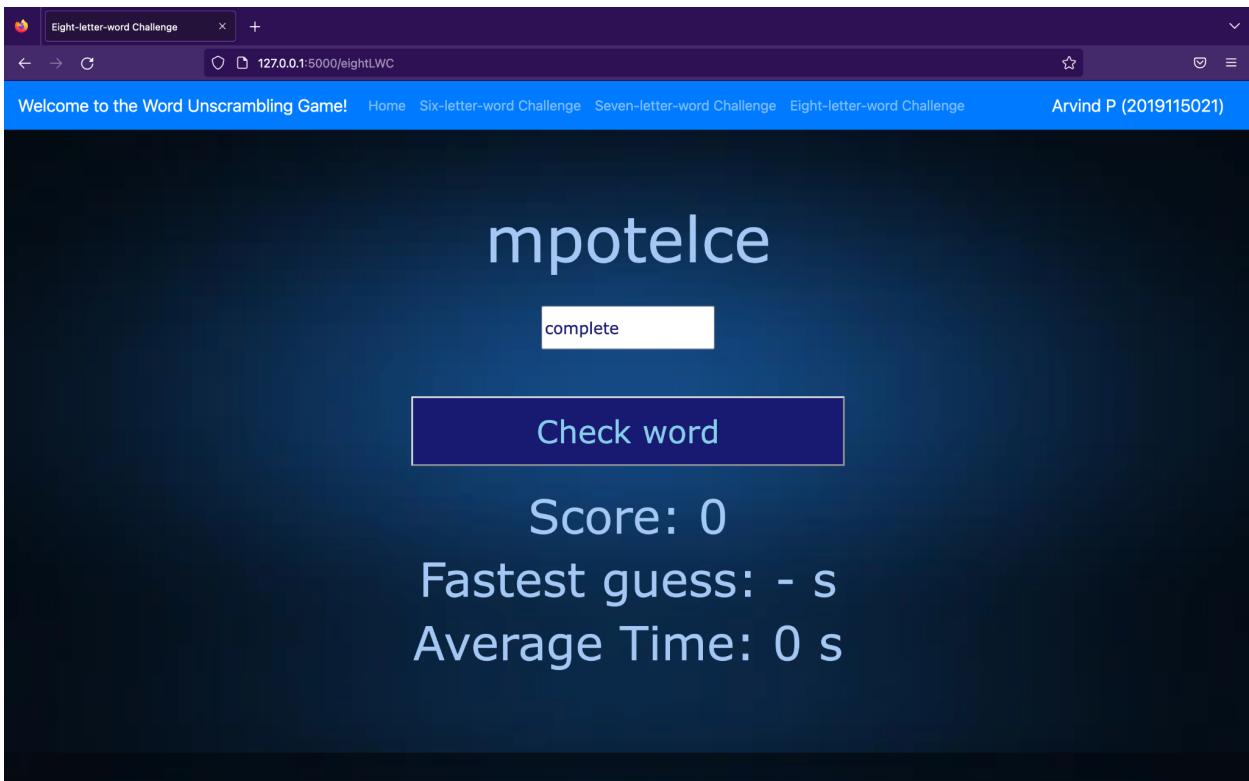
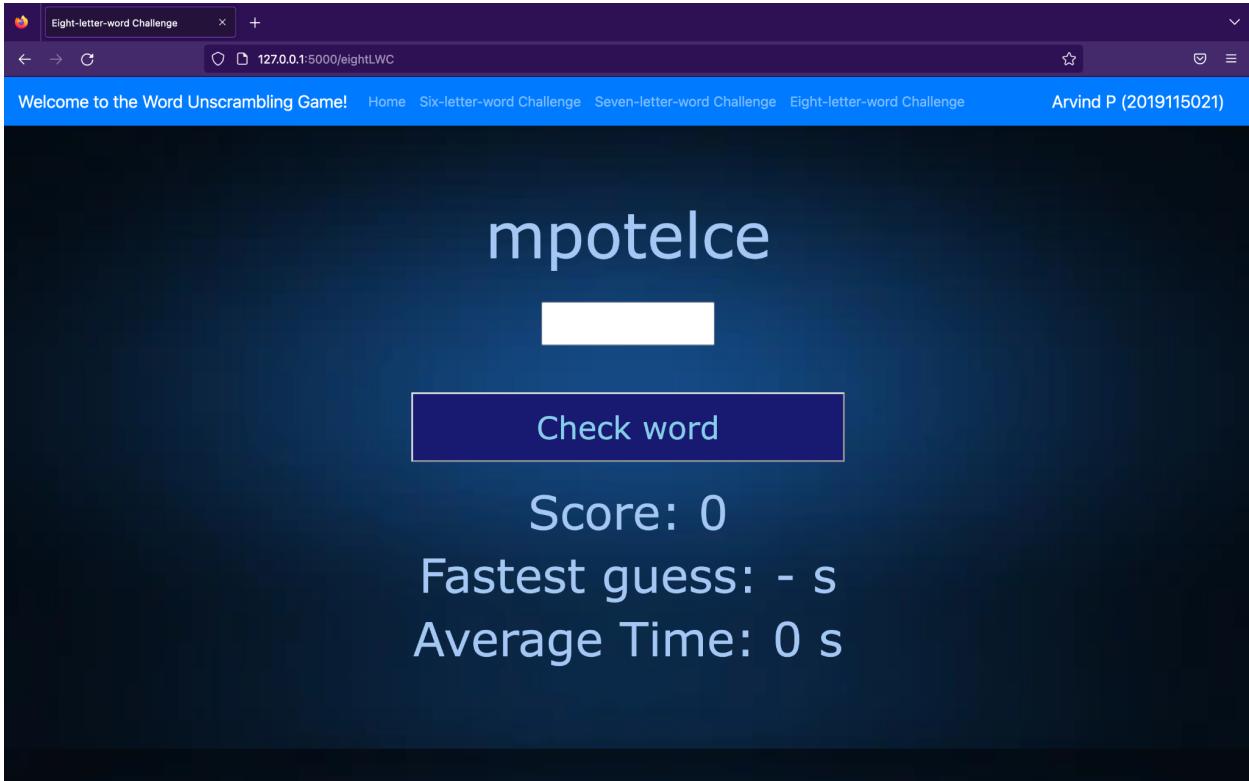
Seven-letter-word challenge:

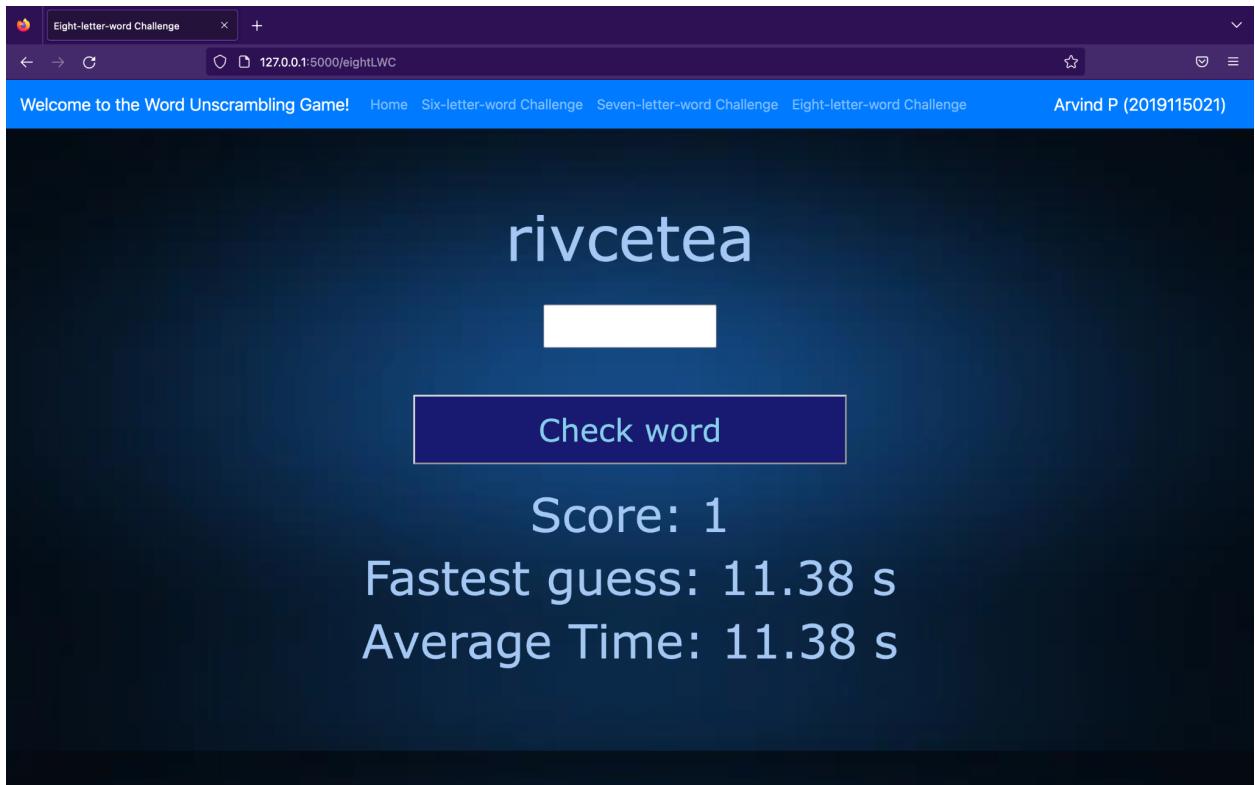


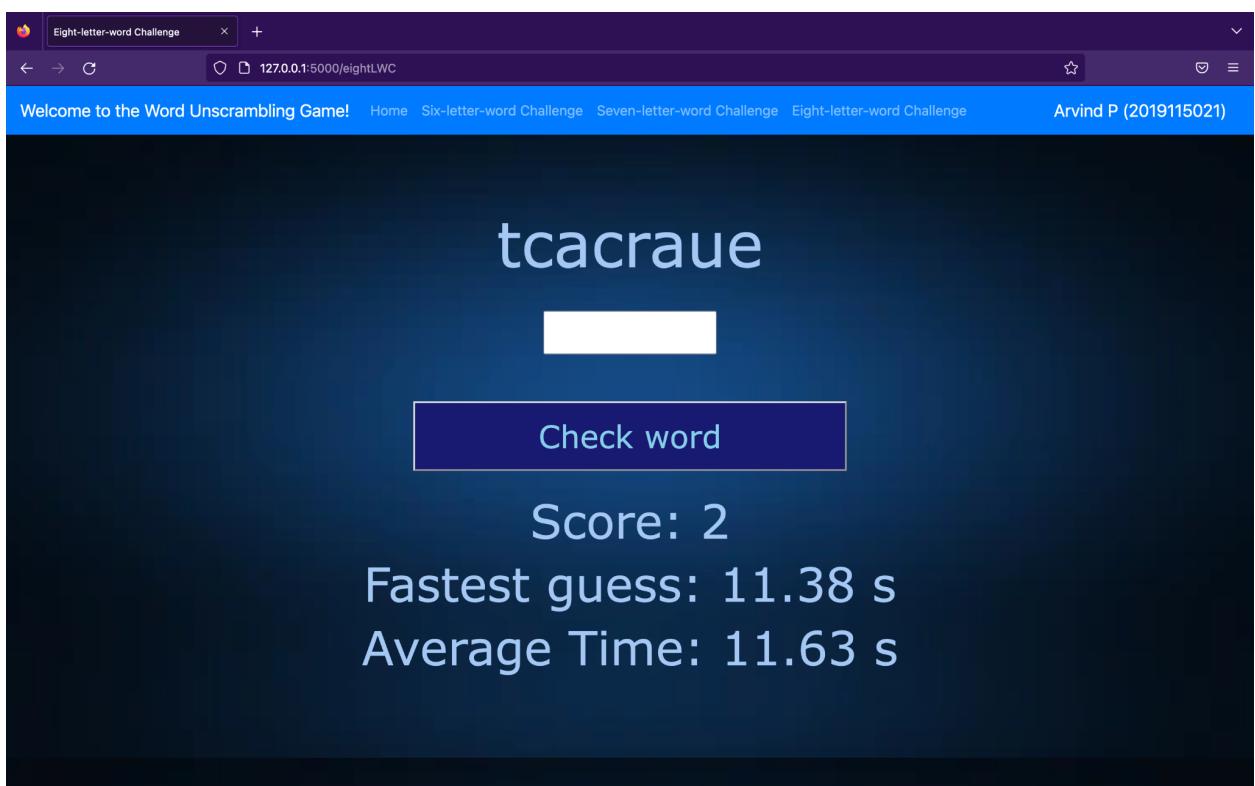
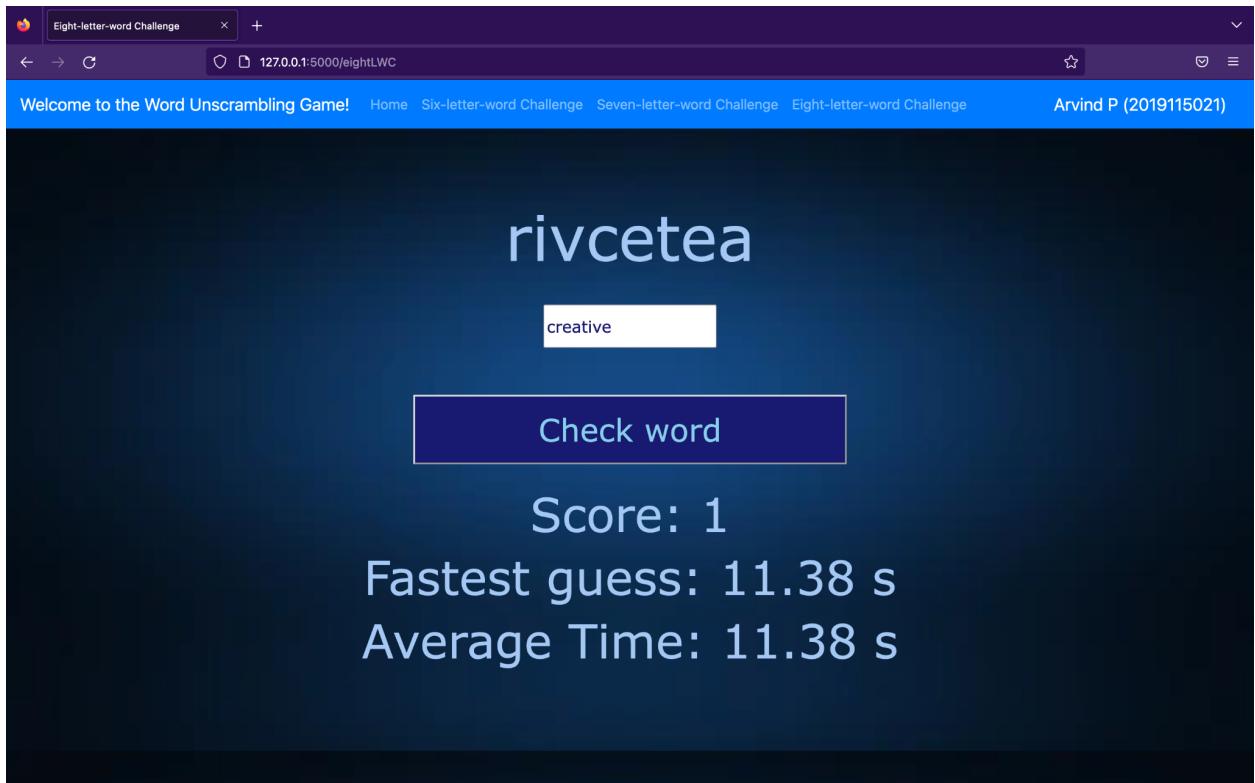




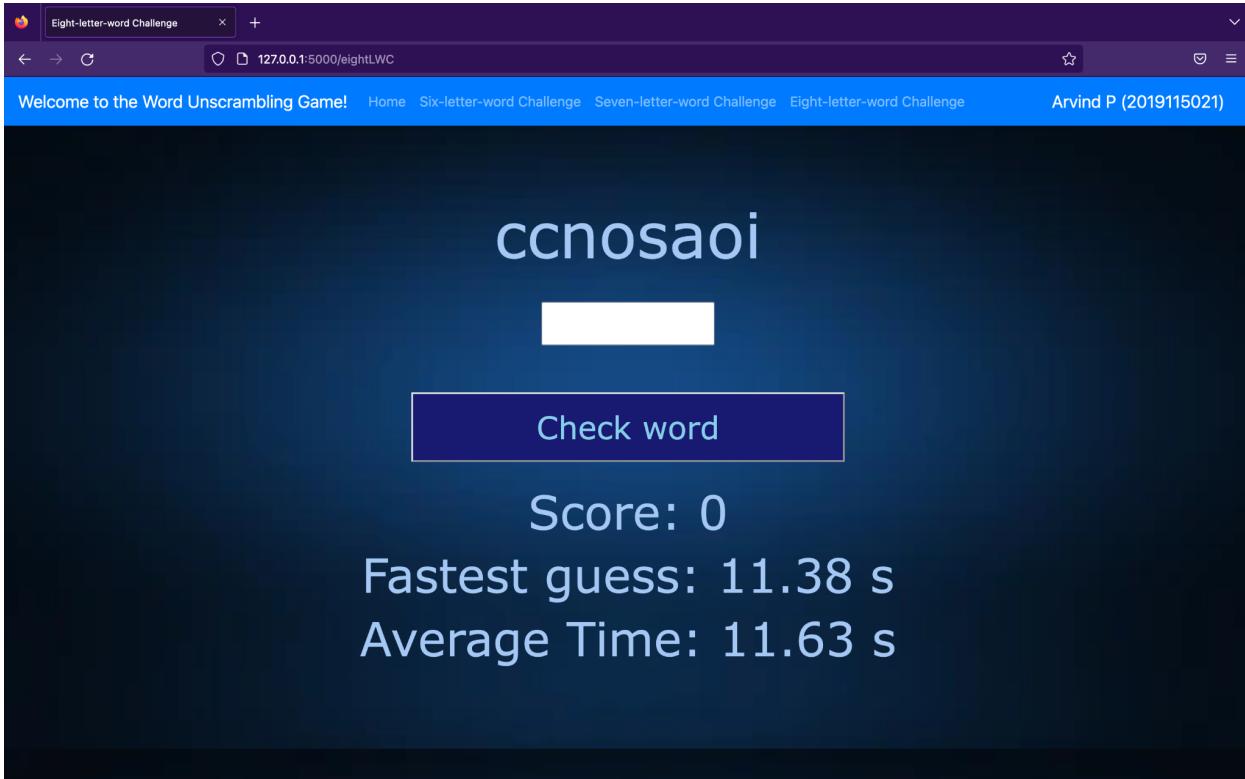
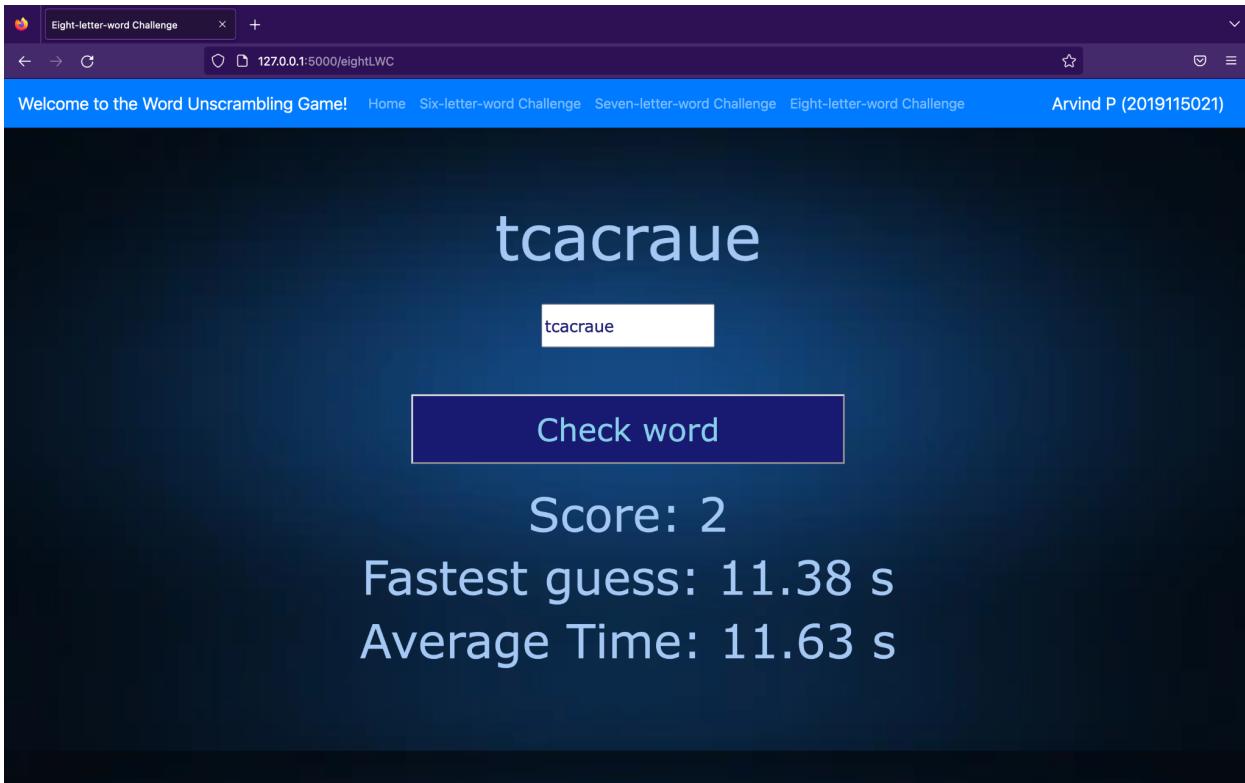
Eight-letter-word challenge:



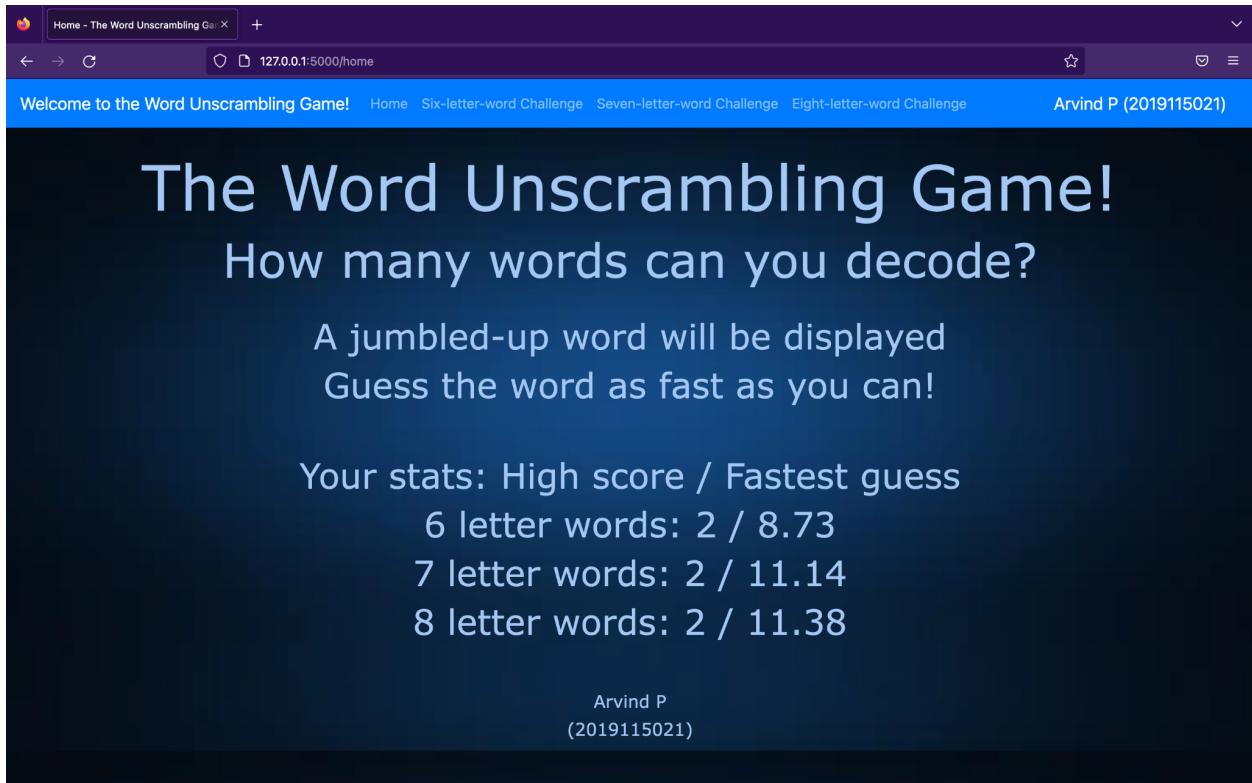




Incorrect word guess:



Updated home page:



4. Codes:

app.py:

```
from flask import Flask, redirect, url_for, render_template, request
from pymongo import MongoClient
import random
from timeit import default_timer as timer

client = MongoClient('localhost', 27017)
db = client.FSP

collection6 = db.wordList6
collection7 = db.wordList7
collection8 = db.wordList8
```

```

class Counter:
    count6 = 0
    count7 = 0
    count8 = 0
    time6 = 0
    time7 = 0
    time8 = 0
    highscore6 = 0
    highscore7 = 0
    highscore8 = 0
    besttime6 = "--"
    besttime7 = "--"
    besttime8 = "--"
    avgtime6 = 0
    avgtime7 = 0
    avgtime8 = 0

app = Flask(__name__)

# HOME PAGE
@app.route("/")
@app.route("/home", methods=["POST", "GET"])
def home():
    return render_template("home.html", highscore6 = Counter.highscore6, highscore7 =
Counter.highscore7, highscore8 = Counter.highscore8, besttime6 = Counter.besttime6,
besttime7 = Counter.besttime7, besttime8 = Counter.besttime8)

# SIX LETTER WORD CHALLENGE
@app.route("/sixLWC", methods=["POST", "GET"])
def sixLW():
    if request.method == "GET":
        cursor = collection6.find()
        no_of_records = collection6.count_documents({})
        global word

        index = 0
        n = random.randint(0, no_of_records - 1)
        scrambled_word = ""

        for record in cursor:
            if index == n:
                word = record["word"]

```

```

        break
    index += 1

scrambled_word = ''.join(random.sample(word, len(word)))
while scrambled_word == word:
    scrambled_word = ''.join(random.sample(word, len(word)))

global start
start = timer()
return render_template("sixLWC.html", scrambled_word = scrambled_word, count6 =
Counter.count6, besttime6 = Counter.besttime6, avgtime6 = Counter.avgtime6)
else:
    global end
    end = timer()
    guessed_word = request.form["guessed_word"]

if guessed_word == word:
    Counter.time6 = round(end - start, 2)
    if Counter.avgtime6 == "-":
        Counter.avgtime6 = round(Counter.time6, 2)
    else:
        Counter.avgtime6 = round((Counter.avgtime6*Counter.count6 +
Counter.time6) / (Counter.count6 + 1), 2)
    if Counter.besttime6 == "-" or Counter.besttime6 > Counter.time6:
        Counter.besttime6 = Counter.time6
    Counter.count6 += 1
    if Counter.count6 > Counter.highscore6:
        Counter.highscore6 = Counter.count6
else:
    Counter.count6 = 0

return redirect(url_for("sixLW"))

# SEVEN LETTER WORD CHALLENGE
@app.route("/sevenLWC", methods=["POST", "GET"])
def sevenLW():
    if request.method == "GET":
        cursor = collection7.find()
        no_of_records = collection7.count_documents({})
        global word

    index = 0

```

```

n = random.randint(0, no_of_records - 1)
scrambled_word = ""

for record in cursor:
    if index == n:
        word = record["word"]
        break
    index += 1

scrambled_word = ''.join(random.sample(word, len(word)))
while scrambled_word == word:
    scrambled_word = ''.join(random.sample(word, len(word)))

global start
start = timer()
return render_template("sevenLWC.html", scrambled_word = scrambled_word, count7
= Counter.count7, besttime7 = Counter.besttime7, avgtime7 = Counter.avgtime7)
else:
    global end
    end = timer()
    guessed_word = request.form["guessed_word"]

    if guessed_word == word:
        Counter.time7 = round(end - start, 2)
        if Counter.avgtime7 == "-":
            Counter.avgtime7 = round(Counter.time6, 2)
        else:
            Counter.avgtime7 = round((Counter.avgtime7*Counter.count7 +
Counter.time7) / (Counter.count7 + 1), 2)
        if Counter.besttime7 == "-" or Counter.besttime7 > Counter.time7:
            Counter.besttime7 = Counter.time7
        Counter.count7 += 1
        if Counter.count7 > Counter.highscore7:
            Counter.highscore7 = Counter.count7
    else:
        Counter.count7 = 0

    return redirect(url_for("sevenLW"))

# EIGHT LETTER WORD CHALLENGE
@app.route("/eightLWC", methods=["POST", "GET"])
def eightLW():

```

```

if request.method == "GET":
    cursor = collection8.find()
    no_of_records = collection8.count_documents({})
    global word

    index = 0
    n = random.randint(0, no_of_records - 1)
    scrambled_word = ""

    for record in cursor:
        if index == n:
            word = record["word"]
            break
        index += 1

    scrambled_word = ''.join(random.sample(word, len(word)))
    while scrambled_word == word:
        scrambled_word = ''.join(random.sample(word, len(word)))

    global start
    start = timer()
    return render_template("eightLWC.html", scrambled_word = scrambled_word, count8
= Counter.count8, besttime8 = Counter.besttime8, avgtime8 = Counter.avgtime8)
else:
    global end
    end = timer()
    guessed_word = request.form["guessed_word"]

    if guessed_word == word:
        Counter.time8 = round(end - start, 2)
        if Counter.avgtime8 == "-":
            Counter.avgtime8 = round(Counter.time6, 2)
        else:
            Counter.avgtime8 = round((Counter.avgtime8*Counter.count8 +
Counter.time8) / (Counter.count8 + 1), 2)
        if Counter.besttime8 == "-" or Counter.besttime8 > Counter.time8:
            Counter.besttime8 = Counter.time8
        Counter.count8 += 1
        if Counter.count8 > Counter.highscore8:
            Counter.highscore8 = Counter.count8
    else:
        Counter.count8 = 0

```

```
    return redirect(url_for("eightLW"))

if __name__ == "__main__":
    app.run(debug=True)
```

Templates:

base.html:

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
    integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
    crossorigin="anonymous">
    <link rel="stylesheet" href="../static/styles.css">
    <title>{% block title %}{% endblock %}</title>
</head>
<body style="background-image: url('static/img/background.jpg'); background-size:
cover">

    <nav class="navbar navbar-expand-lg navbar-dark bg-primary"
    style="background-color: #e3f2fd;">
        <a class="navbar-brand" href="#">Welcome to the Word Unscrambling Game!</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse"
        data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
        aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item">
                    <a class="nav-link" href="home">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="sixLWC">Six-letter-word Challenge</a>
                </li>
                <li class="nav-item">
```

```

        <a class="nav-link" href="sevenLWC">Seven-letter-word Challenge</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="eightLWC">Eight-letter-word Challenge</a>
    </li>
</ul>
</div>
<a class="navbar-brand" href="#">Arvind P (2019115021)</a>
</nav>

{% block content %}

{% endblock %}

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzbzo5smXKp4YfRvH+8abTE1Pi6jizo"
crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>

</body>
</html>

```

home.html:

```

{% extends "base.html" %}

{% block title %}Home - The Word Unscrambling Game{% endblock %}

{% block content %}

<br/>
<h1 class = "h1"> The Word Unscrambling Game! </h1>
<h2 class = "h2a" > How many words can you decode? </h2> <br/>
<h2 class = "h2b"> A jumbled-up word will be displayed </h2>
<h2 class = "h2b"> Guess the word as fast as you can! </h2>

```

```

<br/> <br/>
<h2 class = "h2b"> Your stats: High score / Fastest guess
<h2 class = "h2b"> 6 letter words: {{ highscore6 }} / {{ besttime6 }} </h2>
<h2 class = "h2b"> 7 letter words: {{ highscore7 }} / {{ besttime7 }} </h2>
<h2 class = "h2b"> 8 letter words: {{ highscore8 }} / {{ besttime8 }} </h2>

<br/> <br/>
<h3 class = "h3"> Arvind P </h3>
<h3 class = "h3"> (2019115021) </h3>

{%- endblock %}

```

sixLWC.html:

```

{% extends "base.html" %}

{% block title %} Six-letter-word Challenge {% endblock %}

{% block content %}

<form action=".//sixLWC" method="POST">

<br/>
<br/>
<br/>
<h1 class = "h1"> {{ scrambled_word }} </h1>
<br/>
<input type="text" class="textbox" name="guessed_word" autocomplete="off"/>
<br/><br/><br/>
<input type="submit" class="bstyle" value="Check word"/>
</form>

<br/>
<h2 class = "h2a"> Score: {{ count6 }} </h2>
<h2 class = "h2a"> Fastest guess: {{ besttime6 }} s </h2>
<h2 class = "h2a"> Average Time: {{ avgtime6 }} s </h2>

{%- endblock %}

```

sevenLWC.html:

```

{% extends "base.html" %}

{% block title %} Seven-letter-word Challenge {% endblock %}

{% block content %}

<form action=".//sevenLWC" method="POST">

<br/>
<br/>

```

```

<br/>
<h1 class = "h1"> {{ scrambled_word }} </h1>
<br/>
<input type="text" class="textbox" name="guessed_word" autocomplete="off"/>
<br/><br/><br/>
<input type="submit" class="bstyle" value="Check word"/>
</form>

<br/>
<h2 class = "h2a"> Score: {{ count7 }} </h2>
<h2 class = "h2a"> Fastest guess: {{ besttime7 }} s </h2>
<h2 class = "h2a"> Average Time: {{ avgtime7 }} s </h2>
{%- endblock %}

```

eightLWC.html:

```

{%- extends "base.html" %}

{%- block title %} Eight-letter-word Challenge {%- endblock %}

{%- block content %}

<form action=".//eightLWC" method="POST">
<br/>
<br/>
<br/>
<h1 class = "h1"> {{ scrambled_word }} </h1>
<br/>
<input type="text" class="textbox" name="guessed_word" autocomplete="off"/>
<br/><br/><br/>
<input type="submit" class="bstyle" value="Check word"/>
</form>

<br/>
<h2 class = "h2a"> Score: {{ count8 }} </h2>
<h2 class = "h2a"> Fastest guess: {{ besttime8 }} s </h2>
<h2 class = "h2a"> Average Time: {{ avgtime8 }} s </h2>
{%- endblock %}

```

Static:

styles.css:

```
.body {  
    background-size: cover;  
    background-position: center;  
    background-attachment: fixed;  
}  
.h1 {  
    font-family: 'Verdana'; font-size: 72px; text-align: center; color: #a5c8f3;  
}  
.h2a {  
    font-family: 'Verdana'; font-size: 54px; text-align: center; color: #a5c8f3;  
}  
.h2b {  
    font-family: 'Verdana'; font-size: 40px; text-align: center; color: #a5c8f3;  
}  
.h3 {  
    font-family: 'Verdana'; font-size: 20px; text-align: center; color: #a5c8f3;  
}  
p, form {  
    font-family: 'Verdana'; font-size: 18px; text-align: center; color: #a5c8f3;  
}  
  
.bstyle {  
    display: block;  
    width :500px;  
    height: 80px;  
    margin-left: auto; margin-right: auto; font-size: 36px; background-color:  
#191970; color: #87CEEB;  
}  
.textbox {  
    height: 50px; width: 200px; font-size: 14pt; color: #191970;  
}
```

5. MongoDB:

FSProj

Collections

Create collection View

Sort by Collection Name

wordList6

Storage size: 20.48 kB Documents: 10 Avg. document size: 39.00 B Indexes: 1 Total index size: 36.86 kB

wordList7

Storage size: 20.48 kB Documents: 10 Avg. document size: 40.00 B Indexes: 1 Total index size: 36.86 kB

wordList8

Storage size: 20.48 kB Documents: 10 Avg. document size: 41.00 B Indexes: 1 Total index size: 36.86 kB

MONGOSH

FSProj

Documents FSP.wordList6

FSP.wordList6

10 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter Type a query: { field: 'value' } Reset Find More Options

ADD DATA EXPORT COLLECTION

1 - 10 of 10

`_id: ObjectId('63845930c81bab8d402192ac')`
`word: "abroad"`

`_id: ObjectId('63845949c81bab8d402192ad')`
`word: "accept"`

`_id: ObjectId('63845957c81bab8d402192ae')`
`word: "supply"`

`_id: ObjectId('6393526716837b806c6578bd')`
`word: "serene"`

`_id: ObjectId('6393528916837b806c6578be')`
`word: "pirate"`

`_id: ObjectId('6393529b16837b806c6578bf')`
`word: "strieve"`

`_id: ObjectId('639352b016837b806c6578c0')`
`word: "saturn"`

`_id: ObjectId('639352d716837b806c6578c1')`
`word: "flight"`

MONGOSH

FSProj ... **Documents** FSP.wordList7 +

My Queries Databases Search

FSP.wordList7

10 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter Type a query: { field: 'value' } Reset Find More Options ▾

ADD DATA EXPORT COLLECTION

1 - 10 of 10 ▶ ▷ ▷ ▷ ▷ ▷ ▷ ▷ ▷ ▷

- `_id: ObjectId('6393535016837b806c6578c6')`
word: "jupiter"
- `_id: ObjectId('6393536216837b806c6578c7')`
word: "balloon"
- `_id: ObjectId('6393537b16837b806c6578c8')`
word: "morning"
- `_id: ObjectId('6393539016837b806c6578c9')`
word: "mineral"
- `_id: ObjectId('639353af16837b806c6578ca')`
word: "stadium"
- `_id: ObjectId('639353d616837b806c6578cc')`
word: "station"
- `_id: ObjectId('639353eb16837b806c6578cd')`
word: "highway"
- `_id: ObjectId('639353f816837b806c6578ce')`
word: "builder"

> MONGOSH ^

FSProj ... **Documents** FSP.wordList8 +

My Queries Databases Search

FSP.wordList8

10 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter Type a query: { field: 'value' } Reset Find More Options ▾

ADD DATA EXPORT COLLECTION

1 - 10 of 10 ▶ ▷ ▷ ▷ ▷ ▷ ▷ ▷ ▷ ▷

- `_id: ObjectId('6393547916837b806c6578d3')`
word: "birthday"
- `_id: ObjectId('6393549716837b806c6578d4')`
word: "concrete"
- `_id: ObjectId('639354b516837b806c6578d5')`
word: "accurate"
- `_id: ObjectId('639354c216837b806c6578d6')`
word: "business"
- `_id: ObjectId('639354d416837b806c6578d7')`
word: "creative"
- `_id: ObjectId('639354e416837b806c6578d8')`
word: "complete"
- `_id: ObjectId('639354fa16837b806c6578d9')`
word: "junction"
- `_id: ObjectId('6393550d16837b806c6578da')`
word: "occasion"

> MONGOSH ^