**Unity Coding Test**

Name: Vinayak Kalaskar

Email Id.: [vinayak1592@gmail.com](mailto:vinayak1592@gmail.com)

**Github Link to all the code:** <https://github.com/vinayak-sk/Unity-Coding>

1. **Section 1: CS Fundamentals**

**Question 1: String Reversal Question:**

**Code =>**

**StringReverse Class:**

Github Link:

[https://github.com/vinayak-sk/Unity-Coding/blob/master/CS Fundamentals/stringReverse.java](https://github.com/vinayak-sk/Unity-Coding/blob/master/CS%20Fundamentals/stringReverse.java)

package Unity;

public class stringReverse {

public static void main(String[] args) {

String input = "Hello World";

stringReverse sr = new stringReverse();

String result = sr.reverseString(input);

System.out.println("input: "+input);

System.out.println("output: " + result);

}

public String reverseString(String input){

if(input == null || input.length() == 0){

return input;

}

char[] inputCharArray = input.toCharArray();

int length = inputCharArray.length;

for(int i = 0, j = (length - 1); i < j; i++, j--){

char temp = inputCharArray[j];

inputCharArray[j] = inputCharArray[i];

inputCharArray[i] = temp;

}

return new String(inputCharArray);

}

}

**StringReverse Unit Tests:**

Github Link:

[https://github.com/vinayak-sk/Unity-Coding/blob/master/CS Fundamentals/StringReverseTest.java](https://github.com/vinayak-sk/Unity-Coding/blob/master/CS%20Fundamentals/StringReverseTest.java)

package Unity;

import org.junit.Test;

import static org.junit.Assert.assertEquals;

import static org.junit.Assert.assertNotEquals;

public class StringReverseTest {

@Test

public void nullCheckTest(){

String str = null;

stringReverse sr = new stringReverse();

assertEquals(sr.reverseString(str), null);

}

@Test

public void lengthZeroCheckTest(){

String str = "";

stringReverse sr = new stringReverse();

assertEquals(sr.reverseString(str), "");

}

@Test

public void stringReverseCheckTest(){

String str = "abc";

stringReverse sr = new stringReverse();

assertEquals(sr.reverseString(str), "cba");

}

@Test

public void stringWithSpacesCheckTest(){

String str = "abc d";

stringReverse sr = new stringReverse();

assertEquals(sr.reverseString(str), "d cba");

}

}

**Question 2: Queue**

**Code =>**

**FIFO Class**

Github Link:

<https://github.com/vinayak-sk/Unity-Coding/blob/master/CS Fundamentals/FIFO.java>

package Unity;

import java.util.Stack;

public class FIFO<T> {

private Stack<T> enqueueStack = new Stack<T> ();

private Stack<T> dequeueStack = new Stack<T> ();

public void enqueue(T insertValue){

enqueueStack.push(insertValue);

}

public T dequeue(){

if(isEmpty())

return null;

if(dequeueStack.isEmpty()){

while(!enqueueStack.isEmpty()){

dequeueStack.push(enqueueStack.pop());

}

}

return dequeueStack.pop();

}

public boolean isEmpty(){

if(enqueueStack.isEmpty() && dequeueStack.isEmpty()){

return true;

}

return false;

}

public int size(){

return enqueueStack.size() + dequeueStack.size();

}

}

**FIFO Unit Tests:**

Github Link:

<https://github.com/vinayak-sk/Unity-Coding/blob/master/CS%20Fundamentals/FIFOTest.java>

package Unity;

import org.junit.Test;

import static org.junit.Assert.assertEquals;

import static org.junit.Assert.assertNotEquals;

public class FIFOTest {

@Test

public void nullValuesCheckTest(){

FIFO<Integer> fifoObject = new FIFO<Integer>();

fifoObject.enqueue(null);

fifoObject.enqueue(null);

assertEquals(fifoObject.dequeue(), null);

}

@Test

public void isEmptyNullValuesCheckTest(){

FIFO<Integer> fifoObject = new FIFO<Integer>();

fifoObject.enqueue(null);

fifoObject.enqueue(null);

assertEquals(fifoObject.isEmpty(), false);

}

@Test

public void dequeueCheckTest(){

FIFO<Integer> fifoObject = new FIFO<Integer>();

fifoObject.enqueue(1);

fifoObject.enqueue(2);

assertEquals(fifoObject.dequeue(), new Integer(1));

}

}

1. **Section 2: Frontend**

**Question 1:**

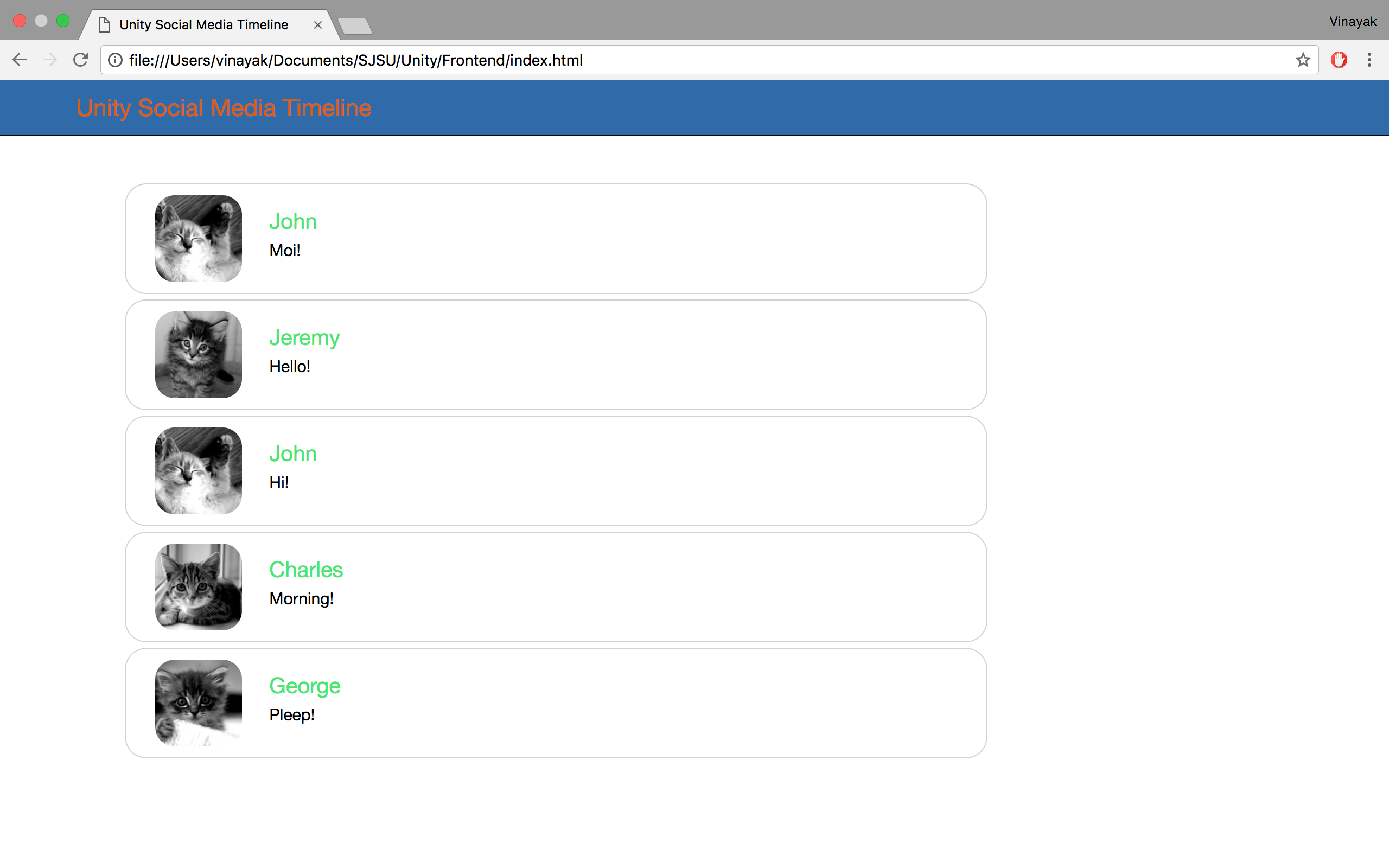
Github Link: <https://github.com/vinayak-sk/Unity-Coding/tree/master/Frontend>

The above link includes all HTML + CSS + JavaScript files required for creating the UI.

Libraries Used:

* Bootstrap for HTML structuring.
* Angular Js for a simple web app.

Screenshot:



**Question 2: Conversation Summaries**

JsFiddle Link: <http://jsfiddle.net/3ghqbh02/50/>

1. **Section 4: Data Processing**

**Question 1: Stats Collector**

JsFiddle Link: <https://jsfiddle.net/y4pchcby/10/>

**Question 2: Grouping Key/Value Pairs**

**MultiValueMap Class:**

package Unity;

import java.util.\*;

public class MultiValueMap {

private static Map<String, List<String>> multiValueMap;

public static boolean add(String key, String value){

if(!multiValueMap.containsKey(key)){

multiValueMap.put(key, new LinkedList<String>());

}

return multiValueMap.get(key).add(value);

}

public static void print(){

for(String key : multiValueMap.keySet()){

System.out.println(key+":");

for(String value : multiValueMap.get(key))

System.out.println(value);

System.out.println();

}

}

}

**MultiValueMap Test Cases:**

package Unity;

import org.junit.Test;

import static org.junit.Assert.assertEquals;

import static org.junit.Assert.assertNotEquals;

public class MultiValuMapTest {

@Test

public void valuesCheckTest(){

MultiValueMap mvp = new MultiValueMap();

mvp.add("1", "1");

mvp.add("1", "2");

mvp.add("1", "4");

mvp.add("2", "1");

mvp.print();

}

@Test

public void nullValuesCheckTest(){

MultiValueMap mvp = new MultiValueMap();

mvp.add(null, null);

mvp.add("1", "2");

mvp.add("1", "4");

mvp.add("2", "1");

mvp.print();

}

}