WEB APPLICATION DEVELOPMENT WITH JAVASCRIPT AND MONGODB

An Industrial Internship Report

submitted by

ANUJ SHUKLA

(18BCE0163)

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

VIT®

Vellore Institute of Technology

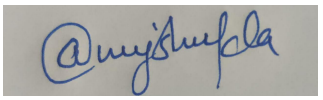
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

OCTOBER 2021

DECLARATION BY THE CANDIDATE

I hereby declare that the Industrial Internship report entitled "**WEB APPLICATION DEVELOPMENT WITH JAVASCRIPT AND MONGODB"** submitted by me to Vellore Institute of Technology, Vellore in partial fulfilment of the requirement for the award of the degree of **B.Tech.** in **Computer Science and Engineering** is a record of bonafide industrial training undertaken by me under the supervision of **Dr. Matthew Yee-King, University of London & Goldsmiths, University of London.** I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.



Name:Anuj Shukla
Reg. Number:18BCE0163

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

BONAFIDE CERTIFICATE

This is to certify that the Industrial Internship report entitled "**WEB APPLICATION DEVELOPMENT WITH JAVASCRIPT AND MONGODB"** submitted by **ANUJ SHUKLA(18BCE0163)** to Vellore Institute of Technology, Vellore in partial fulfilment of the requirement for the award of the degree of **B.Tech.** in **Computer Science and Engineering** is a record of bonafide Industrial Internship undertaken by him/her under my supervision. The training fulfils the requirements as per the regulations of this Institute and in my opinion, meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Signature of the Supervisor

Name>
SUPERVISOR
<Academic Designation>

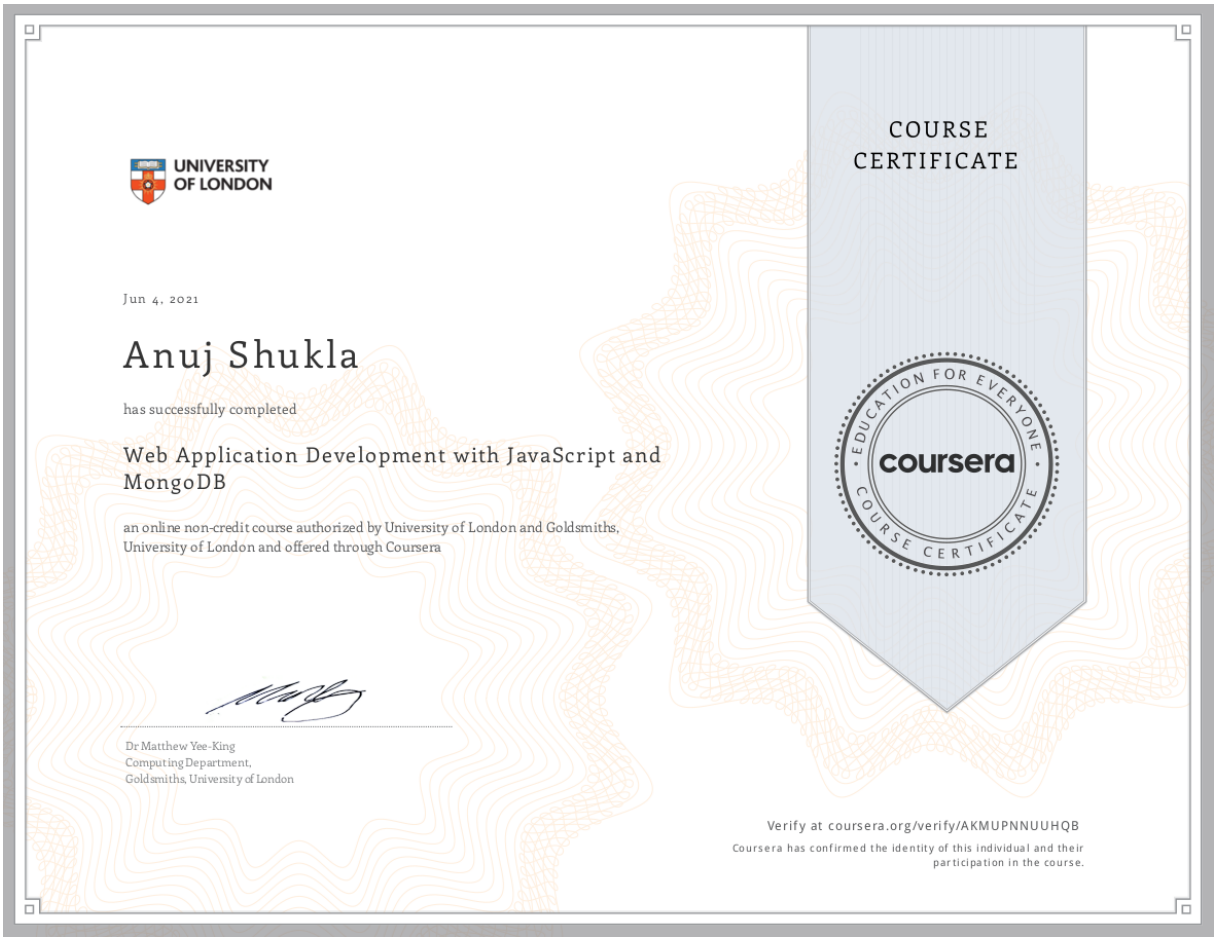
Date:

Date:

<Signature>
Internal Examiner (s)

<Signature>
External Examiner (s)

CERTIFICATE BY THE ORGANIZATION



ACKNOWLEDGEMENT

I would like to add a few words of appreciation for the people who have been a part of this project right from its inception. The writing of this project has been one of the significant academic challenges I have faced and without the support, patience, and guidance of the people involved, this task would not have been completed. It is to them I owe my deepest gratitude.

I would like to thank my mentor Dr. Matthew Yee-King, who always supported me, helping me get results of better quality. It is because of him that I could complete the task assigned to me before the deadlines. His wisdom, knowledge, and commitment to the highest standards inspired and motivated me. Without his insight, support, and energy, this project wouldn't have kick-started and neither would have reached fruitfulness.

Place : Vellore

Anuj Shukla

Date : 16th November 2021

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF ABBREVIATIONS

HTML : Hyper Text Markup Language

JS : JavaScript

UI : User Interface

CSS : Cascading Style Sheets

JSON : JavaScript Object Notation

1. SYNOPSIS

The course I pursued was "Web Application Development with JavaScript and MongoDB" offered by University of London, London and taught by Dr. Matthew Yee-King. The course focuses on learning the basics of reactive web application development. This course, spread over 4 weeks, with a peer-graded assignment and a graded quiz at the end of every week, which mainly focus on the lessons that were covered in that particular module. The objectives of the course are to get started with the web application development.

The course teaches about how meteor framework of JavaScript programming language is effective and how it can be used to write real time applications at scale. The course also goes through the fundamentals of mongoDB. The grading in the course was a peer-review grading system, where we were supposed to code/implement according to the given instructions.

After doing this course, I am now able to write and understand JavaScript code easily. The assignments and in-lecture assignments were extremely beneficial in honing and expanding my skill set. Further, I took it upon myself to explore more and work on a personal project using the knowledge gained in the course. I attempted to create a Space Jumper Web application which is inspired by infamous game doodle jump I played during my childhood. I wrote the web app UI in HTML, CSS and Javascript.

2. ABOUT THE ORGANIZATION – UNIVERSITY OF LONDON & GOLDSMITHS

2.1 Overlay

Goldsmiths, University of London, is a public research university in London, England, specialising in the arts, design, humanities, and social sciences. It is a constituent college of the University of London. It was founded in 1891 as Goldsmiths' Technical and Recreative Institute by the Worshipful Company of Goldsmiths in New Cross, London. It was acquired by the University of London in 1904 and was renamed Goldsmiths' College. The word College was dropped from its branding in 2006, but Goldsmiths' College, with the apostrophe, remains the institution's formal legal name.[3]

Nearly 20% of students come from outside the UK, and 52% of all undergraduates are mature students (aged 21 or over at the start of their studies). Around a third of students at Goldsmiths are postgraduate students.

3. SKILLET BEFORE TRAINING

The skills acquired through the curriculum which helped with the training period are listed as follows

a. Database Management System and MySQL to assist me in my understanding of the framework of a website and the difference between frontend and backend development.

b. Software engineering process and project management skills

c. Ability to analyse a problem, identify and define the computing requirements appropriate to its solution

d. Ability to communicate effectively with a range of audiences.

e. Design and conduct experiments as well as analyse and interpret data.

f. Recognition of the need for and an ability to engage in continuing professional learning (lifelong learning).

g. Ability to use current techniques, skills and tools necessary for computing and engineering practice.

h. Ability to apply mathematical foundations, algorithmic principles and computer science theory in the modelling and design of computer-based systems.

i. Ability to apply design and development principles in the construction of software systems.

Apart from these topics I had basic knowledge of Web Development and had developed a couple of web applications using HTML, CSS and JS. I had a knowledge of programming languages such as Python, JavaScript. Thus, the above-mentioned topics included my skill set at that time.

4. KNOWLEDGE ACQUIRED FROM TRAINING

The course “Web Application Development with JavaScript and MongoDB” consists of 4 different modules. The instructor, Dr. Matthew Yee-King, teaches different aspects of the Go programming language through these modules. The detailed breakdown of each module is given below

4.1 Module 1

This first module is about understanding the bits and pieces of Web Development and meteor programming language. The learnings from this module are described below:

- Operate the mongodb command line client
- Distinguish between valid and invalid returns from database find queries
- Identify reactive data sources in meteor
- Use the Session object to store persistent data
- Describe variable scope
- Use iframes to embed separate web pages

4.2 Module 2

In this module we will learn how to use the core user accounts packages, customise the user accounts UI using third party packages and search for and add packages to an application. We will also look at how to query MongoDB collections from the command line and learn how to control data write access using methods. The learning objectives from this module are as follows:

- Use the core user accounts packages
- Modify the user accounts ui using packages
- Choose packages to add to your project

- Inspect mongodb collections from the command line
- Apply data access controls using methods

4.3 Module 3

In this module we will use bootstrap icons and use the publish and subscribe model to control data read access. We will understand the concept of asynchronous execution and work with template data contexts and helper functions. Finally we will use packages to implement in-place content editing and use complex MongoDB filters. The learning objectives from this module are as follows:

- Use bootstrap icons
- Use publish and subscribe to control data read access
- Describe the concept of asynchronous execution
- Employ mongo data filters

4.4 Module 4

In this module we will create a well organised application using special Meteor folders and we will organise templates into multiple files. We will also learn how to use the iron:router package to create multiple page applications and we will generate data entry forms automatically using SimpleSchema and autoform. Finally, we will validate user data automatically using SimpleSchema. The learning objectives for this module are as follows:

- Reorganize a meteor application using special folders
- Reorganize templates into separate files
- Use the iron:router package to create multiple page applications
- Generate data entry forms automatically using SimpleSchema and autoform
- Use SimpleSchema to validate data automatically

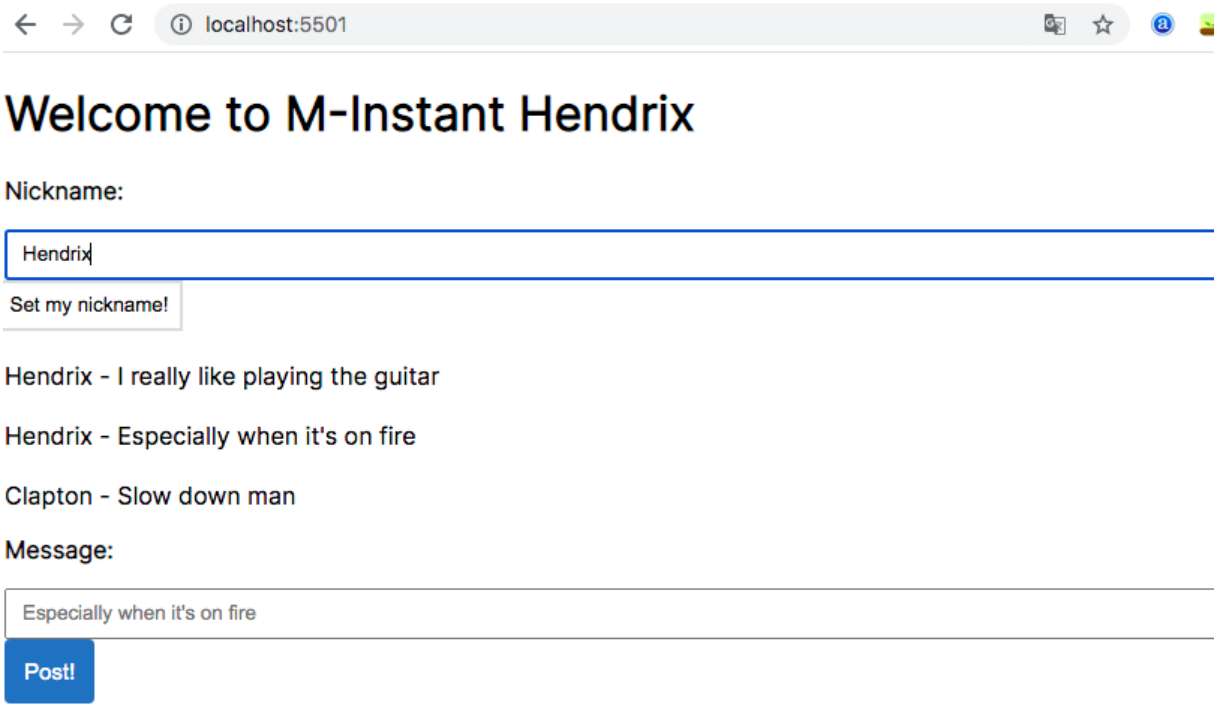
5. APPLICATION OF GAINED KNOWLEDGE

- **Assignment 1**

Q: In this peer assessment you are going to start working on an instant messaging application. You are provided with starter code and you are to make 3 small changes to the application which will make use of the Session variable and meteor's packaging system.

My Submission:

- Please upload a screenshot showing the message list with nicknames at the start of the messages and at the end of the top title. Also, it should show the effect of adding the bootstrap package.



Peer review:

You will see that it shows user nicknames before the messages and it has been styled with bootstrap.

Does the screenshot that you are assessing show the nicknames at the start of the messages?

- 0 points
  No
- **1 point**
  Yes

S1
SS

Does the screenshot look like it has been styled with bootstrap as in the example?

- 0 points
  No
- **1 point**
  Yes

S1
SS

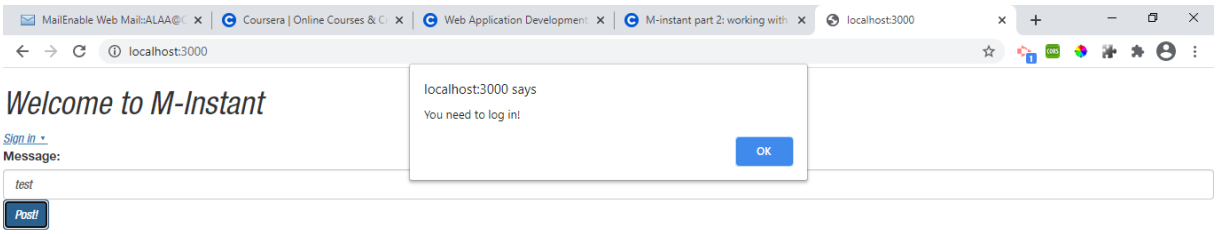
- **Assignment 2**

In this assessment, you will work with user accounts and the mongo command line.

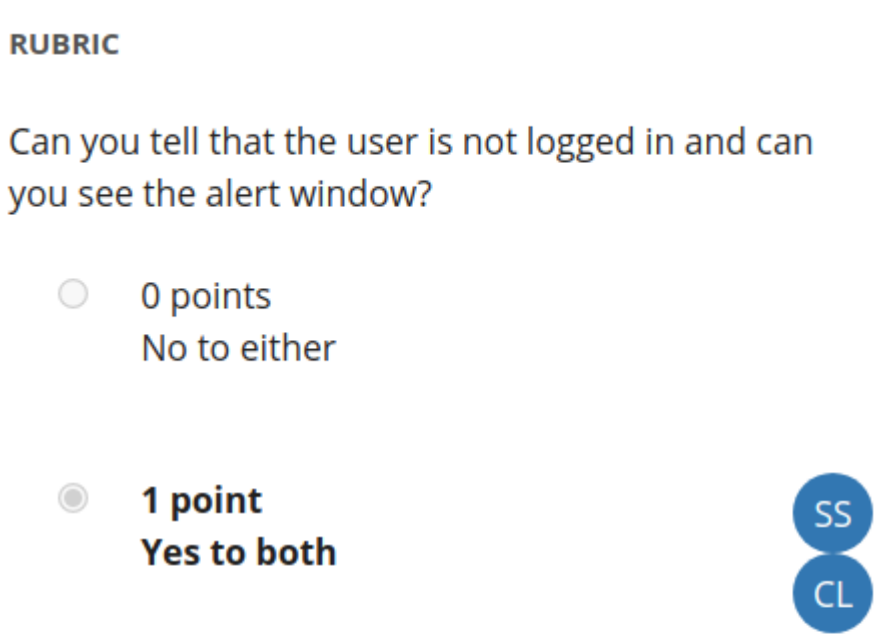
You will make some small changes to the minstant application such that users have to login to post messages and so that their username is used to set the nickname.

My Submission:

- Upload a screenshot showing a non-logged in user receiving an alert message after trying to post a message.



Peer Review:



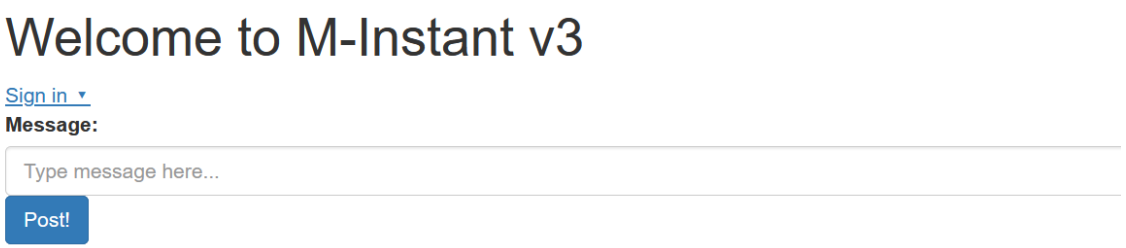
- **Assignment 3**

In this exercise, you are going to work with publish and subscribe so that you can gain control over read access in your project.

You will be assessed based on successfully implementing the code for a publish function and the code for a subscribe call which responds reactively to the user logging in.

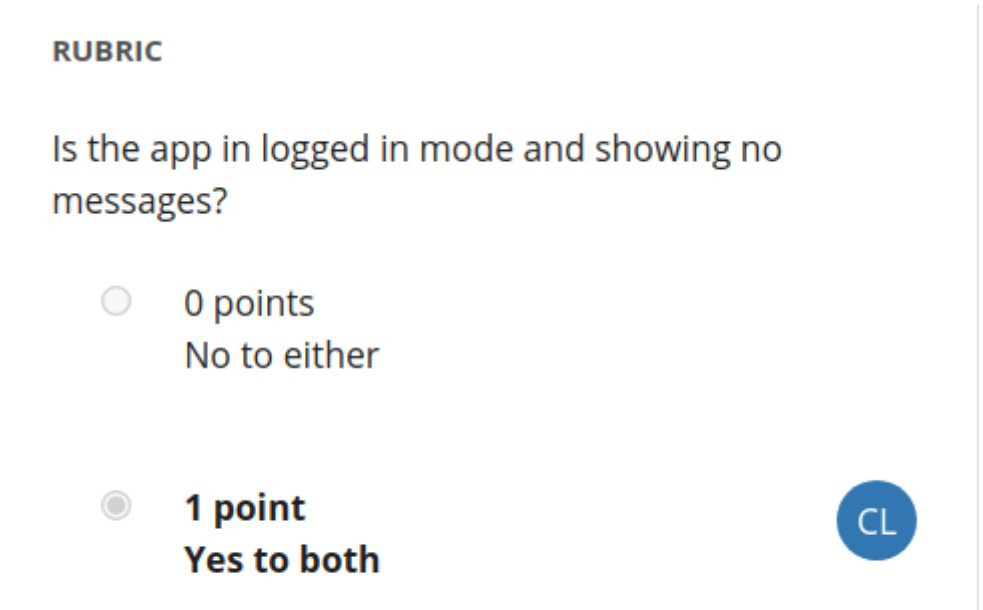
My Submission:

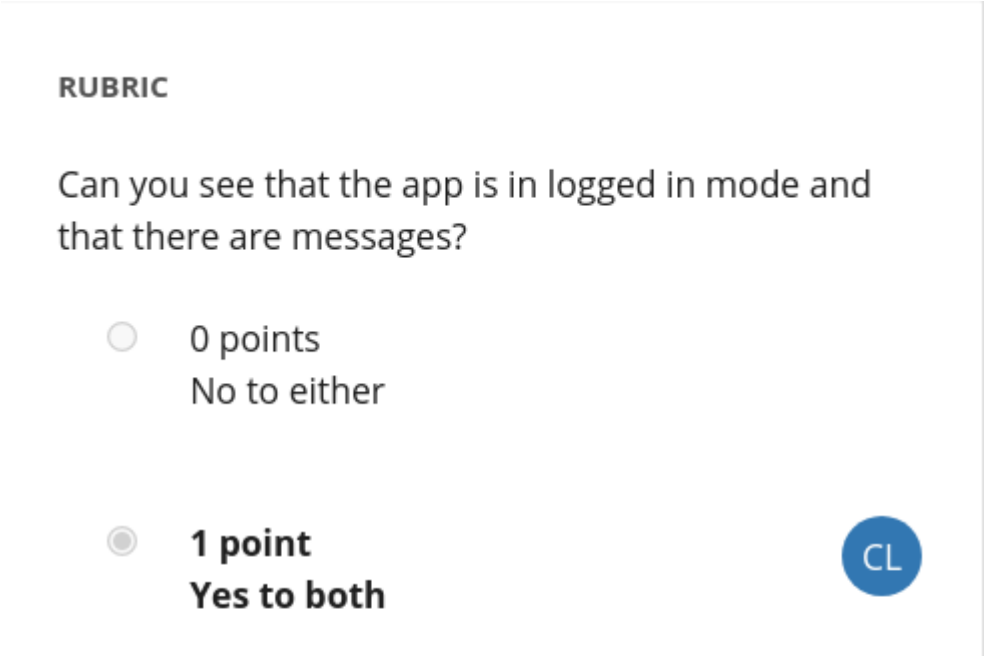
- Upload a screenshot showing the app in logged out mode, with no messages.



- Upload a screenshot showing the app in logged in mode, with messages showing.

Peer Reviews:





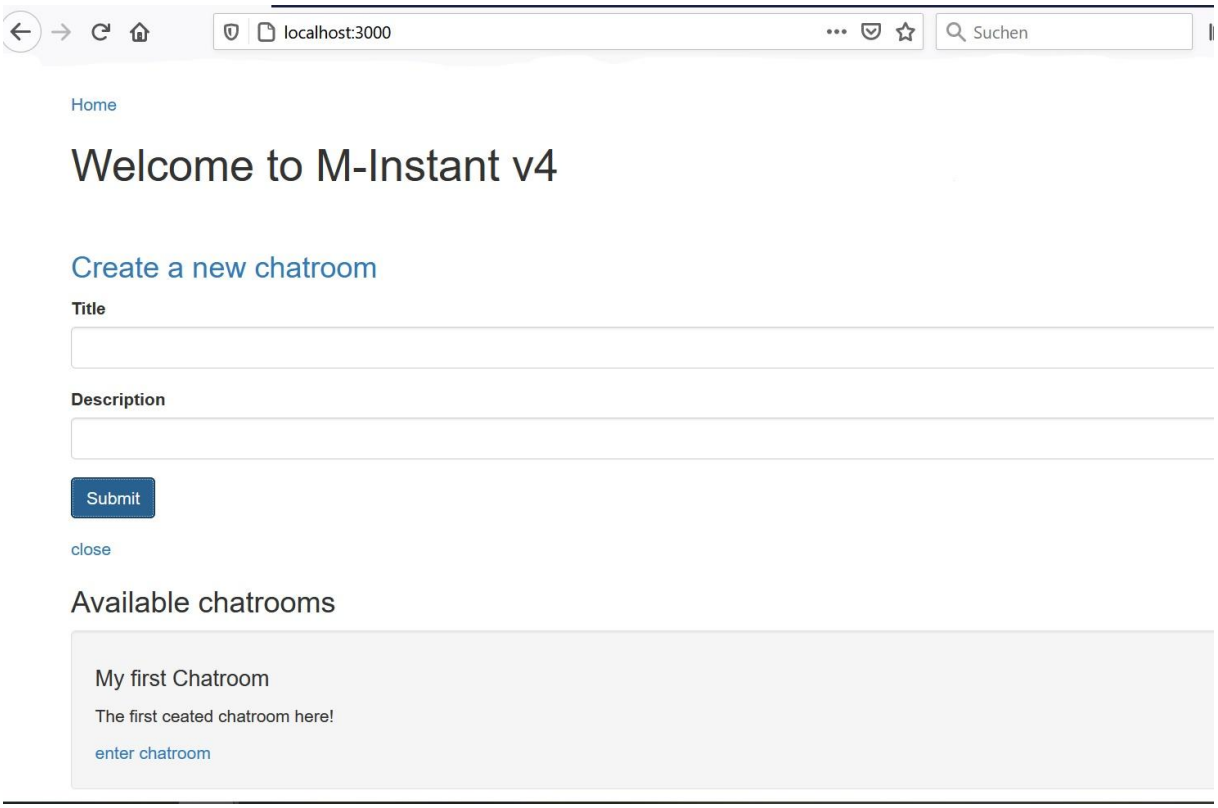
- **Assignment 4**

In this assignment you will add some more functionality to m-instant so it has multiple chatrooms, with the data entry controlled by SimpleSchame and auto generated forms!

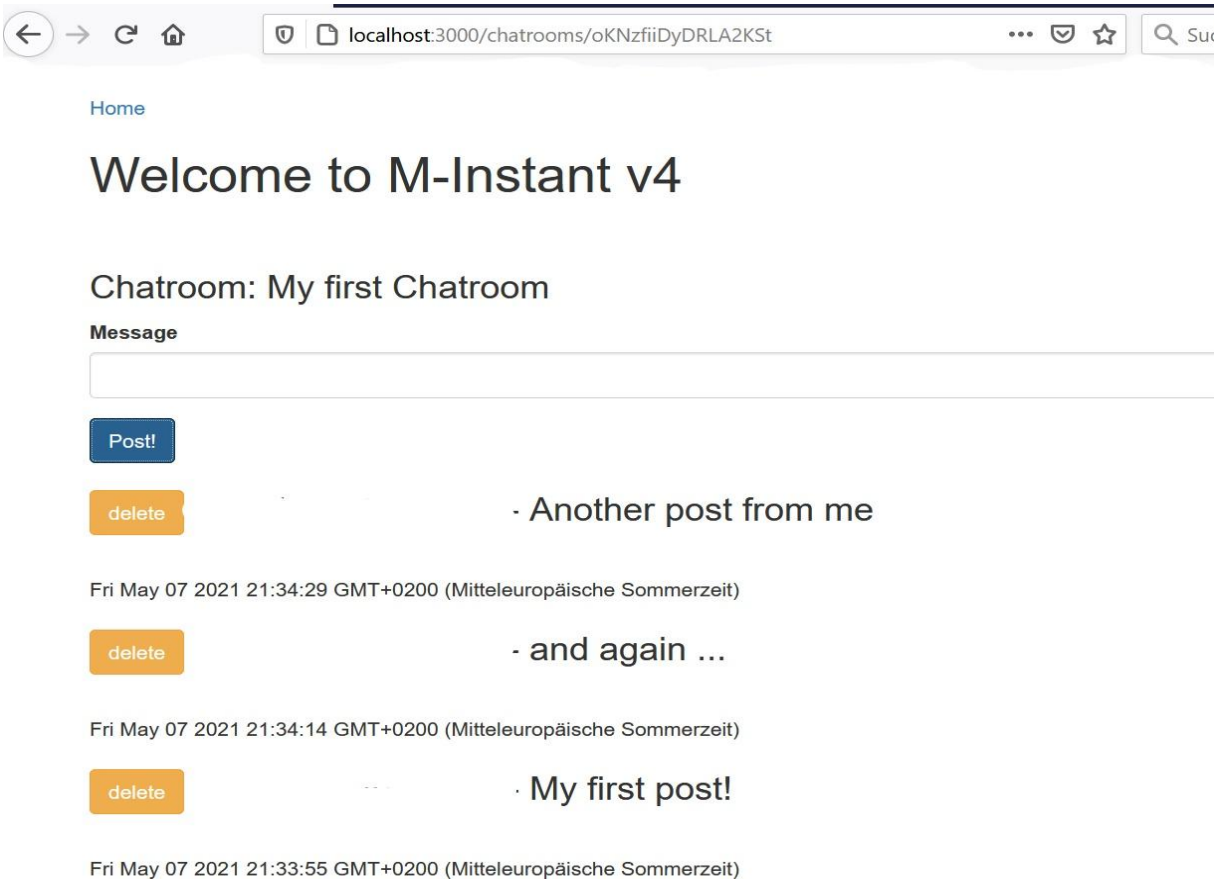
As usual, your work will be assessed based on completing the tasks described and providing evidence that the functionality works using screen shots.

My Submission:

- Upload a screenshot showing the chatroom creation form that is created by autoform, as it appears in your web browser.



- Upload a screenshot showing a list of messages as they appear in a chatroom.



Peer Reviews:

RUBRIC

Can you see title and description fields plus a submit button?

- 0 points
  No to any
- **1 point**
  Yes to all

RUBRIC

Can you see a list of messages plus a chatroom title at the top?

- 0 points
  No to any
- **1 point**
  Yes to all

6. COMPARISON OF COMPETENCY LEVELS

The training has helped develop a lot of new skills. Although I had basic knowledge about some concepts used, the training has helped me apply these skills and helped me better my understanding about these technologies and learn new technologies.

6.1 Before

Before taking the course, I had a basic knowledge of our program's curriculum inclusive of Data Structures and Algorithms, Python, Software Engineering etc. The knowledge imparted to us in the classroom was adequate.

The following was my competency level before taking the course:

- Knowledge about building and implementing small scale projects.
- Basic knowledge of HTML, CSS, Web Development, Python.
- Entry level knowledge of JavaScript.
- Little exposure to concurrency and parallelism.
- Little exposure to usage of git

6.2 After

My competency level after taking the course:

- Advance knowledge of HTML, CSS, Web Development.
- Moderate knowledge of JavaScript.
- Entry level knowledge about real time application development
- Basic knowledge about Package architecture in meteor.
- Moderate knowledge of git.

7. APPENDICES

I further explored Web Development in JavaScript and tried to create my version of Doodle Jump and named it **“Space Jumper”.**

I have deployed the web application on netlify.

The link to the game is : https://spacejumper.netlify.app/

- **app.js**

```
document.addEventListener("DOMContentLoaded", () => {
  const space = document.querySelector(".space");
  const spacecraft = document.createElement("div");
  let spacecraftLeftSpace = 50;
  let startPoint = 150;
  let spacecraftBottomSpace = startPoint;
  let isGameOver = false;
  let platformCount = 5;
  let platforms = [];
  let upTimerId;
  let downTimerId;
  let isJumping = true;
  let isGoingLeft = false;
  let isGoingRight = false;
  let leftTimeId;
  let rightTimeId;
  let score = 0;
  var button;
  var text;
```

```
function createSpaceCraft() {
  space.appendChild(spacecraft);
  spacecraft.classList.add("spacecraft");
  spacecraftLeftSpace = platforms[0].left;
  spacecraft.style.left = spacecraftLeftSpace + "px";
  spacecraft.style.bottom = spacecraftBottomSpace + "px";
}
class Platform {
  constructor(newPlatBottom) {
    this.bottom = newPlatBottom;
    this.left = Math.random() * 315;
    this.visual = document.createElement("div");
    const visual = this.visual;
    visual.classList.add("platform");
    visual.style.left = this.left + "px";
    visual.style.bottom = this.bottom + "px";
    space.appendChild(visual);
  }
}
function createPlatform() {
  for (i = 0; i < platformCount; i++) {
    let platformGap = 600 / platformCount;
    let newPlatBottom = 100 + i * platformGap;
    let newPlatform = new Platform(newPlatBottom);
    platforms.push(newPlatform);
```

```
  }
}

function movePlatforms() {
  if (spacecraftBottomSpace > 200) {
    platforms.forEach((platform) => {
      platform.bottom -= 4;
      let visual = platform.visual;
      visual.style.bottom = platform.bottom + "px";
      if (platform.bottom < 10) {
        let firstPlatform = platforms[0].visual;
        firstPlatform.classList.remove("platform");
        platforms.shift();
        score++;
        playerScore();
        let newPlatform = new Platform(600);
        platforms.push(newPlatform);
      }
    });
  }
}
function jump() {
  clearInterval(downTimerId);
  isJumping = true;
  upTimerId = setInterval(function () {
    spacecraftBottomSpace += 20;
```

```
    spacecraft.style.bottom = spacecraftBottomSpace + "px";
    if (spacecraftBottomSpace > startPoint + 200) {
      fall();
    }
  }, 30);
}
function fall() {
  clearInterval(upTimerId);
  isJumping = false;
  downTimerId = setInterval(function () {
    spacecraftBottomSpace -= 5;
    spacecraft.style.bottom = spacecraftBottomSpace + "px";
    if (spacecraftBottomSpace <= 0) {
      gameOver();
    }
    platforms.forEach((platform) => {
      if (
        spacecraftBottomSpace >= platform.bottom &&
        spacecraftBottomSpace <= platform.bottom + 15 &&
        spacecraftLeftSpace + 60 >= platform.left &&
        spacecraftLeftSpace <= platform.left + 85 &&
        !isJumping
      ) {
        console.log("Landed");
        startPoint = spacecraftBottomSpace;
        jump();
```

```
      }
    });
  }, 30);
}

function gameOver() {
  console.log("Game Over");
  isGameOver = true;
  while (space.firstChild) {
    space.removeChild(space.firstChild);
  }
  space.innerHTML = " <br> <br> <br><br> <br> <br> <br> <br> <br><br>
<br> <br> <br> <br> <br>    Your Score is " + score +"<br> <br>";
  space.innerHTML =space.innerHTML +" <br> <a href="
onClick='window.location.reload()'>Restart </a><br>";
  clearInterval(upTimerId);
  clearInterval(downTimerId);
  clearInterval(leftTimeId);
  clearInterval(rightTimeId);



}

function control(e) {
  if (e.key === "ArrowLeft") {
    moveLeft();
  } else if (e.key === "ArrowRight") {
```

```
    moveRight();
  } else if (e.key === "ArrowUp") {
    moveStraight();
  }
}

function moveLeft() {
  if (isGoingRight) {
    clearInterval(rightTimeId);
    isGoingRight = false;
  }
  isGoingLeft = true;
  leftTimeId = setInterval(function () {
    if (spacecraftLeftSpace >= 0) {
      spacecraftLeftSpace -= 5;
      spacecraft.style.left = spacecraftLeftSpace + "px";
    } else moveRight();
  }, 30);
}
document.querySelector(".left").addEventListener("click", moveLeft)
//  document.querySelector(".up").addEventListener("click", moveStraight)
document.querySelector(".right").addEventListener("click", moveRight)

function moveRight() {
  if (isGoingLeft) {
    clearInterval(leftTimeId);
```

```
    isGoingLeft = false;
  }
  isGoingRight = true;
  rightTimeId = setInterval(function () {
    if (spacecraftLeftSpace <= 340) {
      spacecraftLeftSpace += 5;
      spacecraft.style.left = spacecraftLeftSpace + "px";
    }
  }, 30);
}

function moveStraight() {
  isGoingLeft = false;
  isGoingRight = false;
  clearInterval(rightTimeId);
  clearInterval(leftTimeId);
}


function start() {
  if (!isGameOver) {
    createPlatform();
    createSpaceCraft();
    setInterval(movePlatforms, 30);
    jump();
    document.addEventListener("keyup", control);
```

```
  }
 }
 //attach a start button
 start();
 function playerScore() {

  const playerScore = "Your Score is" + score++;
  const displayScore = document.querySelector(".score");

 }
});
```

- **index.html**

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Space Jumper Made By Anuj</title>
  <link rel="stylesheet" href="style.css" />
  <script src="app.js"></script>
 </head>
 <body>
  <div class="galaxy">
   <div class="stars"></div>
```

```
      <div class="selection">
        <h1 style="color: orangered;">
          Play this game on</h1> <br>
          <a href="comp.html" class="comp">Computer</a> <br>
          <a href="mob.html" class="mob">Mobile</a>

      </div>
    </div>
  </body>
</html>
```

style.css

```
body {
  margin: 0;
  padding: 0;
}
.space {
  width: 600px;
  height: 800px;
  background-color: black;
  position: relative;
  color: orangered;
  text-align: center;
  text-decoration: solid;
  font-weight: bolder;
```

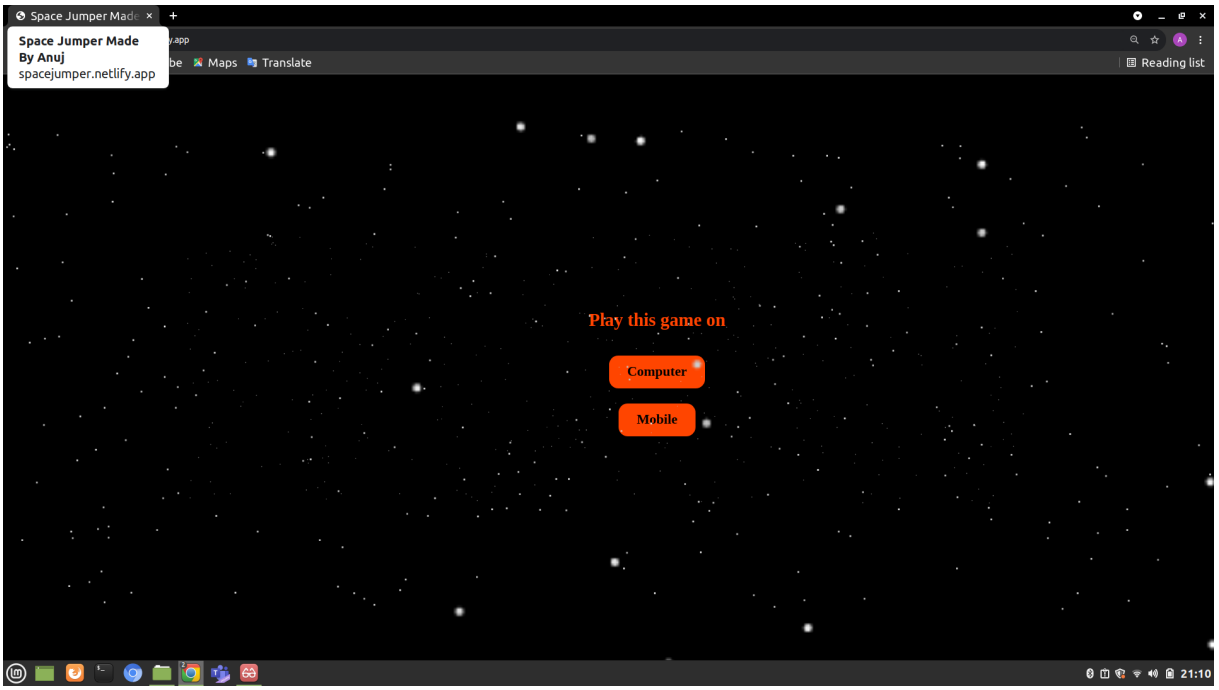
```
  font-size: xx-large;
  display: flex;
  margin-left: 31vw;
  align-items: center;
  opacity: 0.7;
  display: flex;
  flex-wrap: wrap;
  flex-direction: column;
}
@media only screen and (max-width: 600px) {
  .space {
    margin-left: 6vw;
    font-size: large;
    display: flex;
    flex-wrap: wrap;
    flex-direction: column;
    align-items: center;
    align-content: center;
    font-display: center;
    text-align: center;
    padding-left: 0vw;
    position: absolute;
    top: 50%;
    left: 50%;
    -ms-transform: translate(-50%, -50%);
    transform: translate(-50%, -50%);
```

```
  }

  a {
    background-color: orangered;
    border: none;
    color: black;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 4px 2px;
    cursor: pointer;
    font-weight: bolder;
    font-size: x-large;
    border-radius: 1rem;
  }
}
.spacecraft {
  width: 60px;
  height: 80px;
  background-image: url("moon.png");
  border-radius: 50%;
  background-size: cover;
  position: absolute;
  /* opacity: 0.9; */
```

```
}
.selection {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  margin-top: 38vh;
  margin-left: 8vw;
}
a {
  background-color: orangered;
  border: none;
  color: black;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
  font-weight: bolder;
  font-size: x-large;
  border-radius: 1rem;
}
.arrows {
  display: flex;
```

```
  flex-direction: row;
  align-items: flex-end;
  padding: 5vw;
  margin-top: 60vh;
  margin-left: 10vw;
}

.platform {
  width: 85px;
  height: 15px;
  background-color: blue;
  position: absolute;
}
.galaxy-header {
  overflow: hidden;
  position: absolute;
  width: 100%;
  height: 100vh;
  -webkit-perspective: 340px;
  perspective: 340px;
  opacity: 0.7;
}
```

Output:



Link to the project: https://github.com/Atypical-introvert/Space-Jumper