

BCDV1008 - Full Stack III

Chat Application

Due Tues, April 7th^t – 25% of Final Grade

System Requirements

Contact: **Mike Denton**

Date: **Feb 1st, 2020**

Version #: **1.1**

1 Objective

This document contains a specification of the course assignment. It is a task where students practice skills to build a full stack web application using the backend technologies Node.js and MongoDB offered by the MERN Stack. This task will also include, working together in a group developing the project, plan, manage and coordinate development activities, to be done effectively to a deadline.

2 Teams

Teams will consist of one to two students (with two being the target number).

3 Backend Server

The backed server will include Node.js. We will be using Socket.io for the Web Socket communication. Any Node packages built-in or 3rd party found on the NPM registry can be used.

4 Database

We will be using NOSQL Database to persist the data. We will be using the cloud hosted MongoDB on Mongo DB Cloud Atlas <https://www.mongodb.com/cloud/atlas>

5 User Interface

The UI design will be HTML5/CSS3 with ES6 (JQuery can be used, but optional).

6 Specification

The application for the project will be based on one of either the **Chat or Game** application, using Socket.io and optionally phaser.io (<https://phaser.io/>)

7 Socket.io

Implement a Chat application or Game application using Socket.io.

Socket application requirements


































- **listen for when a new socket is created**
 - save when a socket connection is created in history
- **listen for when a new user is joins a room**
 - echo to client that they have connected (via emit)
 - echo to everyone in the room that a person has connected
 - save to **chat history** the a new socket connection is made with timestamp (MongoDb)
 - save to **chat history** the username and room joined (MongoDb)
- **listen for client input (chat message or game position)**
 - update the room of the new input
- **listen for switching room**
 - leave the room and join a new room
 - echo to the user the new room connected
 - send a message to the old room that the user has left
 - update to the new room a new user has joined
- **listen for the user disconnect**
 - update the list of users in the chat the user has disconnected
 - save in **event log** user disconnected with timestamp (MongoDb)

8 Mongoose and MongoDB

Build models and schemas with mongoose middleware to store data in mongoDb

1. Store data in mongoDb using mlabs..
2. Build model and schema to save user chat history
3. Build model and scheme to save socket.io events.
4. Write a mongoose query to retrieve all user chat history
5. Write a mongoose query to retrieve all user chat history by room name
6. Write a mongoose query to retrieve all event logs

[Event Log could be similar to the window event logs you could have a Type fields to capture the type of socket Event \(CONNECTION, DISCONNECT, JOINED, ERROR\) with a Date, Time, User, maybe and Event or PPID field.](#)

Type	Date	Time	Source	Category	Event	User	Comp...
 Error	10/3/2006	5:23:44 PM	Userenv	None	1085	SYSTEM	NBBH...
 Warning	10/3/2006	5:23:44 PM	SceCli	None	1202	N/A	NBBH...
 Information	10/3/2006	3:33:29 PM	SCHEMA1_PROGRESS	DATASE...	2690	N/A	NBBH...
 Information	10/3/2006	3:33:29 PM	SCHEMA1_PROGRESS	DATASE...	6490	N/A	NBBH...
 Information	10/3/2006	3:33:29 PM	SCHEMA1_PROGRESS	DATASE...	6489	N/A	NBBH...
 Information	10/3/2006	3:33:29 PM	SCHEMA1_PROGRESS	DATASE...	0	N/A	NBBH...
 Information	10/3/2006	3:33:29 PM	SCHEMA1_PROGRESS	DATASE...	2689	N/A	NBBH...
 Information	10/3/2006	3:33:29 PM	SCHEMA1_PROGRESS	DATASE...	5067	N/A	NBBH...
 Information	10/3/2006	3:33:22 PM	ORACLE_TEST_PRO...	PROGRE...	12699	N/A	NBBH...
 Information	10/3/2006	3:33:22 PM	ORACLE_TEST_PRO...	PROGRE...	13547	N/A	NBBH...
 Information	10/3/2006	3:33:22 PM	ORACLE_TEST_PRO...	PROGRE...	7161	N/A	NBBH...
 Information	10/3/2006	3:33:22 PM	ORACLE_TEST_PRO...	PROGRE...	5326	N/A	NBBH...
 Information	10/3/2006	3:33:22 PM	ORACLE_TEST_PRO...	PROGRE...	451	N/A	NBBH...
 Error	10/3/2006	3:24:39 PM	Userenv	None	1085	SYSTEM	NBBH...
 Warning	10/3/2006	3:24:39 PM	SceCli	None	1202	N/A	NBBH...
 Information	10/3/2006	3:14:57 PM	SPORTS2_PROGRESS	PROGRE...	12699	N/A	NBBH...
 Information	10/3/2006	3:14:57 PM	SPORTS2_PROGRESS	PROGRE...	13547	N/A	NBBH...
 Information	10/3/2006	3:14:57 PM	SPORTS2_PROGRESS	PROGRE...	7161	N/A	NBBH...
 Information	10/3/2006	3:14:57 PM	SPORTS2_PROGRESS	PROGRE...	5326	N/A	NBBH...
 Information	10/3/2006	3:14:57 PM	SPORTS2_PROGRESS	PROGRE...	451	N/A	NBBH...
 Error	10/3/2006	1:49:34 PM	Userenv	None	1085	SYSTEM	NBBH...
 Warning	10/3/2006	1:49:34 PM	SceCli	None	1202	N/A	NBBH...
 Information	10/3/2006	1:01:56 PM	SPORTS2_PROGRESS	PROGRE...	334	N/A	NBBH...
 Information	10/3/2006	1:01:56 PM	SPORTSORA_PROGR...	PROGRE...	334	N/A	NBBH...
 Information	10/3/2006	1:01:56 PM	TESTORACLE1_PRO...	PROGRE...	334	N/A	NBBH...
 Information	10/3/2006	1:01:56 PM	TESTSCHEMA_PROG...	PROGRE...	334	N/A	NBBH...
 Information	10/3/2006	12:57:50 PM	ORACLE_TEST_PRO...	PROGRE...	334	N/A	NBBH...
 Information	10/3/2006	12:15:18 PM	PROGRESS	DATASE...	2690	N/A	NBBH...
 Information	10/3/2006	12:15:18 PM	PROGRESS	DATASE...	6490	N/A	NBBH...
 Information	10/3/2006	12:15:18 PM	PROGRESS	DATASE...	6489	N/A	NBBH...
 Information	10/3/2006	12:15:18 PM	PROGRESS	DATASE...	0	N/A	NBBH...
 Information	10/3/2006	12:15:18 PM	PROGRESS	DATASE...	2689	N/A	NBBH...
 Information	10/3/2006	12:15:18 PM	PROGRESS	DATASE...	5067	N/A	NBBH...

[Example of Windows Event Log.](#)

[The User Chat History table is more for tracking the chat message sent with Date, Time stamps, along with the sender and receiver users, the chat room/namespace etc.](#)

id	sender	receiver	message
10	kevin	alex	this is a message from kevin
11	anthony	alex	this is a message from anthony
12	kevin	alex	last message from kevin
13	anthony	alex	last message from anthony
14	alex	anthony	hi anthony

Example of Chat History (Date and Time columns should be included)

10 Node.js

Build a web server that and use the middleware required for mongodb

1. Create the web server, mongoose and socket.io connections.

Submission

1. The project code submission is via GitHub Class repository. Please include in your student id folder
2. Include a README file with the project that includes the following:
 - The names and student number of all the members of the tea
 - Instructions for installing or running the project.

Specification	Percentage
Backend Server (Node.js)	30%
Implementing Mongoose	30%
Implementing Socket.io on Client and Server	35%
Clean Code and Clarity	5%

11 Bonus

Bonus marks will be awarded for the following completed items. Note: this is only applicable where the project mark is less than 100%

Specification	Percentage
UI/UX - uses frameworks ie. Bootstrap, React etc.	5%
Deployment to Cloud Server (Heroku, Digital Ocean)	5%