

# Mintbean Take-Home Test

## Full Stack JavaScript

### Introduction

Welcome to the Mintbean take-home test. This is an opportunity for you to provide potential employers with a practical sample of your work.

### Instructions

#### Timelines

If a deadline has been mandated by Mintbean, please follow it.

#### Submitting source code

If no submission instructions have been received, please contact us immediately.

You must send your source code at the end of this test as a .zip, .rar or .7z file. Exact instructions will have been included in your email. **Please do NOT send us .gitignored files such as node\_modules, cache folders, etc.**

#### Architecture

Your project's architecture is mostly up to you. However, the following architectural requirements must be followed:

1. Frontend and backend must be cleanly separated projects
2. Frontend must be written in a single-page-application framework. (React, Angular, Vue, Backbone, Ember, etc.)
3. Backend must serve data to the frontend via RESTful APIs.

### Requirements

Implement the following user stories in a full-stack JavaScript project. You may use any external library, framework, architectural pattern or dependency you'd like. Persistence can be done in any SQL or NoSQL database.

Build a simple chatroom.

1. Unregistered users can register with `username` and a hashed `password`
2. Users can login and logout.
3. Users can chat inside a single room.
4. Users CANNOT create a new room, there is only a single chatroom.

5. Users' messages display the username, the message, and the timestamp in ``yyyy-mm-dd hh:mm`` format.
6. Messages should appear in chronological order.
7. Users' own messages are highlighted with a light green background.
8. Users can "like" each others' messages by clicking on a "like" button that appears on the message.
9. Users can receive messages by refreshing the full page using the browser's refresh button.

(Continued...)

## Scope limits

The following are **out of scope** for this project:

1. "Forgot Password"
2. Styling

## Bonus points

The following is not important to attempt. But if you have extra time, you may attempt it.

1. New messages appear as soon as they are sent by a user. (You can use any technique you like to make this work)
2. Nice UX/UI

## Example screenshot

You may build the application using whatever styling you'd like, as long as it meets the above requirements. However, here is a sample screenshot of one possible implementation of the application.

# Mintbean Chatroom

Hello, Zack. [Logout](#)

Submit

**RITA**

Happy new year, everyone!

2020-01-01 00:59

[Like](#)

**TOMMY**

Hey guys, hope I'm not too late to the party. Happy new decade!

2020-01-01 00:11

[Like](#)

**JASON**

Happy new year!

2020-01-01 00:10

*(Liked)*

**ZACK**

HAPPY NEW YEAR WOOOOO 2020!!

2020-01-01 00:06

**KIMBERLY**

Happy new year, Jason!