

# Fiverr's Front-End Test

## Goal

Develop a fully functional chat application.

The test will focus on your HTML, CSS and JavaScript skills.

You are provided with 3 assets to create said application, all of which are detailed in their own section:

- Visual UI mockup
- Behaviour specs
- Prebuilt SDK
- Prebuilt CSS classes

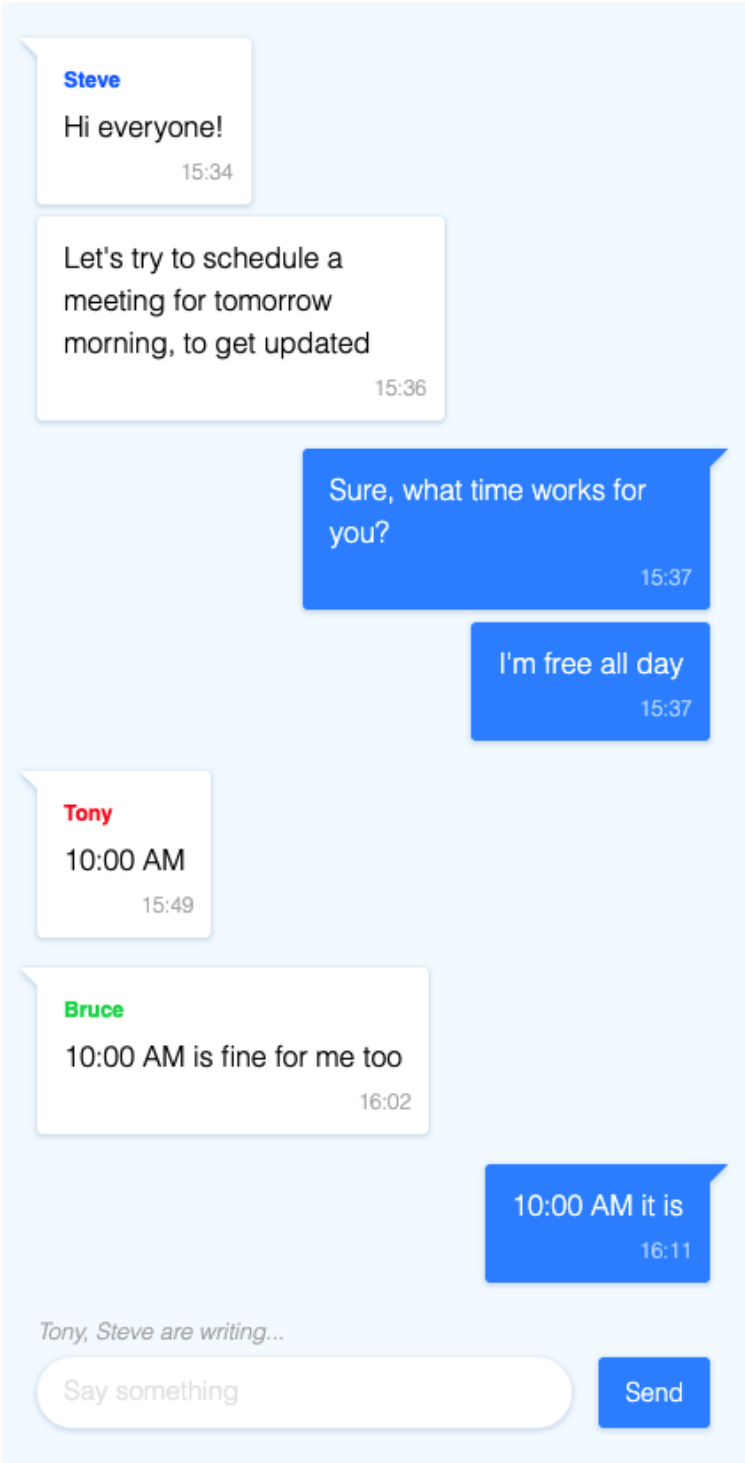
You have **2 hours** to complete this task, without using any external tools or libraries.

## General Guidelines

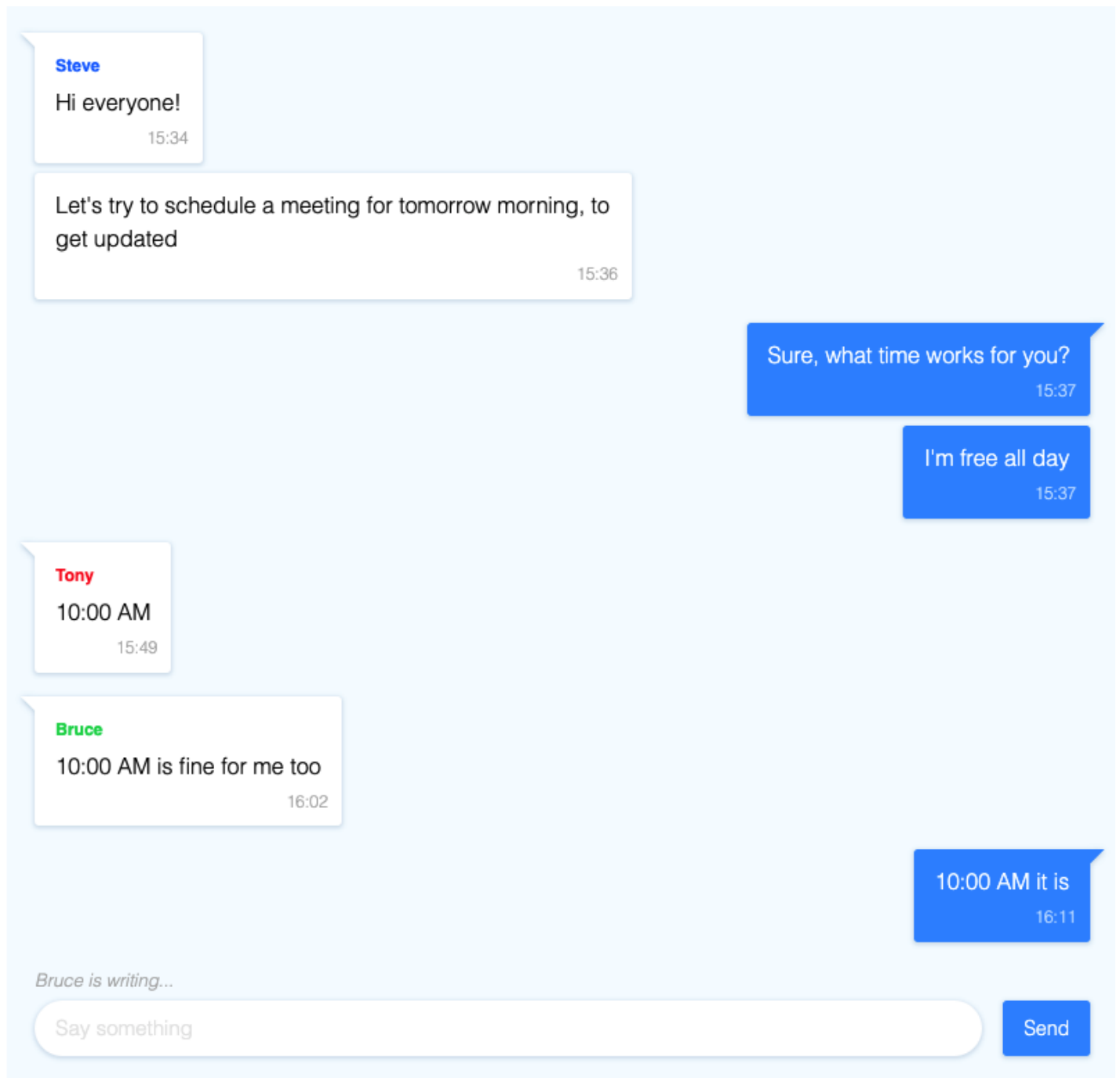
- Your application should be mobile responsive (pay attention to different sizes between the 2 mockups).
- Pay attention to your code structure - We'd like to see a logical separation between functionality and data objects, as well as generic and elegant functions.
- Use relevant, consistent naming.
- Write clean, modern code (HTML5, CSS3, ES2019).

# Mockups

## Mobile



## Desktop



## Expected Behaviour

- The user should be able to send new messages using the controls below
- The user should be able to see incoming messages
- Messages should be grouped by user, with only the first message displaying the user's name and an arrow
- Messages should display the hour and minute in which they were sent
- The chat should be scrollable
- The input and send button should be sticky
- Messages shouldn't overlap with the input and send button
- The user should be able to see which other people are typing in the chat room (without displaying himself).  
A user is considered typing for 8 seconds after he last typed

# SDK

You are provided with some basic prebuilt SDK you can use to send messages and respond to incoming messages.

All chat SDK functionality is under the namespace `window.Chat`.

## **sendMessage(message)**

Sends a message to the chat room.

### **Arguments**

- `message (String)` - The message's content.

## **onMessage(callback)**

Registers a callback function to the send message event, that will be triggered when a message is received in the chat (including the user's messages).

**Only one callback can be registered at a given time.**

The callback function receives one argument - a **Message** (detailed below)

### **Arguments**

- `callback (Function)` - The callback function to run. Receives one argument: a **Message** object.

## **Message**

A schema defining each message's structure.

- `content (String)` - The message's content.
- `user (String)` - The message's sender name. The current user's name is always `"Me"`.
- `timestamp (Date)` - The message's send time.
- `id (String)` - The message's unique id.

## **onTyping(callback)**

Registers a callback function to the typing event, that will be triggered when a user in the chatroom is typing (including the current user).

**Only one callback can be registered at a given time.**

The callback function receives one argument - a username.

### **Arguments**

- `callback (Function)` - The callback function to run. Receives one argument: a username.

# CSS Classes

`.message`

Renders the base of the message (background, shadow and font color).

**.message.self**

Renders the message with the user's color scheme.

**.message.with-arrow**

Renders the message with the arrow.