# DOODLE BUDDY

DANNY / OVIEH / SARA

# WHAT IS DOODLE BUDDY?

- A multi-player game in which one user draws an image based on a prompt and the other user must guess what it is!

*Think Pictionary…*

# MOTIVATION FOR DEVELOPMENT

- **Create an interactive application centered around a new technology**

  - **WebSockets**

    - Open a persistent connection the client and server

      - This allows for an interactive communication session

    - API allows the client to send messages to the server and receive event-driven responses

      - Client and server can send data at any time
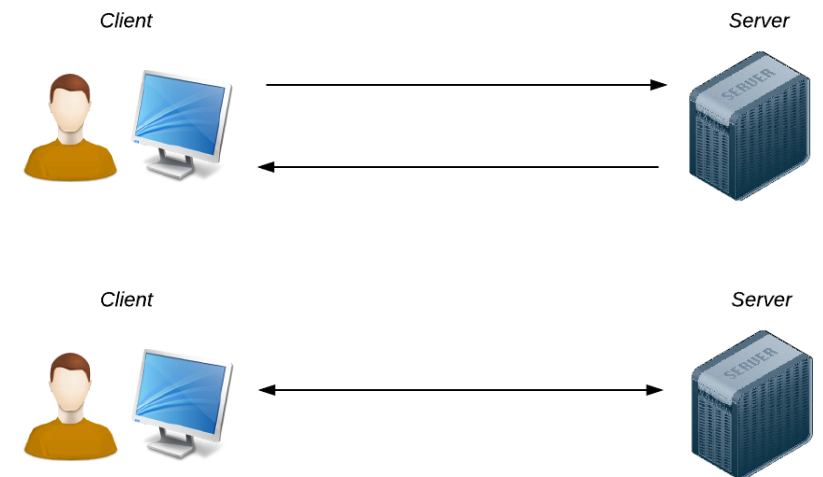
# MOTIVATION FOR DEVELOPMENT

- **Create an interactive application centered around a new technology**

  – **Socket.IO**
    - JavaScript library for working with WebSockets
    - Separate client-side (browser) and server-side (node.js) libraries

  – **Why use for Doodle Buddy?**
    - ❖ Broadcast information to multiple sockets
    - ❖ Store data associated with each client
    - ❖ Asynchronous input/output
    - Lets users interact through chat and canvas
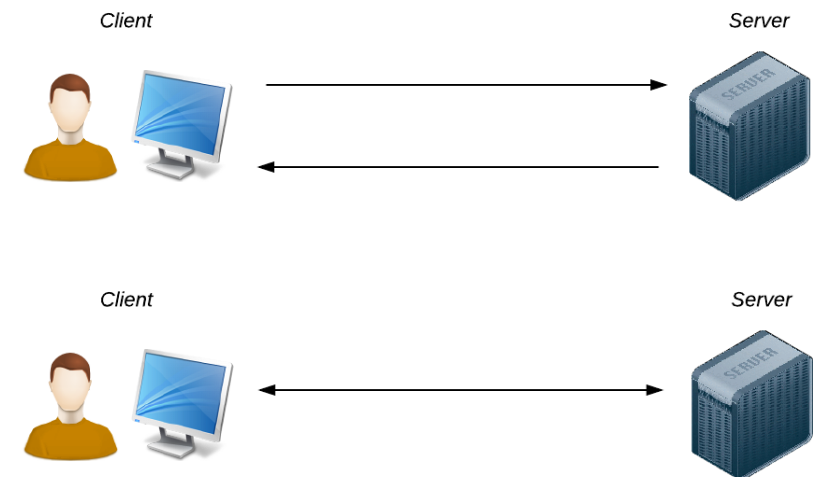
Client                                         Server

Client                                         Server

- body-parser
- chance
- express
- express-handlebars
- materialize-css
- morgan
- mysql
- mysql2
- nodemon
- npm
- sequelize
- socket.io
- webpack

# TECHNOLOGIES USED

/*

*take a look inside our package.json file for dependencies and devDependencies*

*/

# TECHNOLOGIES USED

- body-parser
- chance
- express
- express-handlebars
- **materialize-css**
- morgan
- mysql
- mysql2
- nodemon
- npm
- sequelize
- socket.io
- webpack

/*

*CSS library,*

*designed by Google*

*/

## TECHNOLOGIES USED

- body-parser
- chance
- express
- express-handlebars
- materialize-css
- **morgan**
- mysql
- mysql2
- nodemon
- npm
- sequelize
- socket.io
- webpack

/*

*logger middleware,*

*logs requests to console,*

*helpful for debugging in node*

*/

## TECHNOLOGIES USED

- body-parser
- chance
- express
- express-handlebars
- materialize-css
- morgan
- mysql
- mysql2
- **nodemon**
- npm
- sequelize
- socket.io
- webpack

*/*

*listens for file changes*

*and establishes new*

*server connection each*

*time,*

*productivity++*

**/*

- body-parser
- chance
- express
- express-handlebars
- materialize-css
- morgan
- mysql
- mysql2
- nodemon
- npm
- sequelize
- **socket.io**
- webpack

## TECHNOLOGIES USED

*/*

*JavaScript library for real-time web applications,*

*uses persistent*

*connection to make*

*chat/canvas viewable to*

*both users*

*/*

# TECHNOLOGIES USED

- body-parser
- chance
- express
- express-handlebars
- materialize-css
- morgan
- mysql
- mysql2
- nodemon
- npm
- sequelize
- socket.io
- **webpack**

/*

*static module bundler,*

*packages files into into*

*few static assets for use*

*in browser (client-side)*

*/

# DESIGN PROCESS

- Independent research for new technologies

- Created an initial mock up to determine functional pieces of main webpage

- Create new content on different branches in repository
  - *Merge branches into master to combine work and check viability of all pieces*
    - *canvas, front-end, database, players, users, socket, etc…*

- Slack and class for communication and planning

# DEMONSTRATION

# DIRECTIONS FOR FUTURE DEVELOPMENT

❖ **Scalability**

- Turn-taking and gameplay mechanics currently limited to 2 users
- Expand for larger chatroom and collaborative gameplay

❖ **Utilize user data to create `buddies`**

- Akin to how Words With Friends allows users to start new games with known players
    - Sharing user id or game id

❖ **Saving images**

- Allow players to capture their images and post to social media
- Create image gallery on landing page