# IDEA and USERS What problem(s) does your app solve? How does it solve those problems? Who is your target user? How much experience do they have with technology? Help businesses communicate easier through: - Screen Sharing - Audio Conference - Video Conference - File Transfer

### **FEATURES**

Write use cases for your app. IE - As a user I can store a list of my favorite food. Use color to separate features by 'Minimum Viable Product' and 'Bonus features'

- Team lead can create a team
- email invite to team members
- Screen share?

# VIEW What views do you need to create to meet each feature in your app? How will the user get to each view? Revisit this regularly. Simplify each time. Focus on the user. Make a connection to the controller behind the view. Draw each view and map out how the user will get to each of them. Login Header/Menu Area team C Chat Header/Menu Area create team and add team members Header/Menu Area add team Create End points

## CONTROLLERS

Create a controller for each wireframe you made. Make a connection to each endpoint it needs to use to get its data.

Login Component: - Register

- Passport-Local (Axios) - Return user

Team View:

- User - Get all team users Get team chats

- Admin - Delete/Rename team

- Delete/Add user Socket.io for who is online

#### Chats:

- Submit chat (Socket.io) - Get chats (Socket.io)

Team Creation: - New team

 Add team members Invite members to team

Invite New Member: Add member

Invite new members to team

#### **ENDPOINTS**

List the url, REST method, and a sample of the data being sent or received for every endpoint you need.

Make a connection to the parts of the schema you need to access from each endpoint to return or save the data needed.

#### User:

Create user

- Update profile Delete user

 Login (and with socket.io) Logout (and with socket.io)

#### Team:

- Create team
- Delete team Update team profile
- Get team info
- Add member to team

- Remove member from team

#### Chat:

- Create (post) new chat - Read chats from team

### SCHEMA

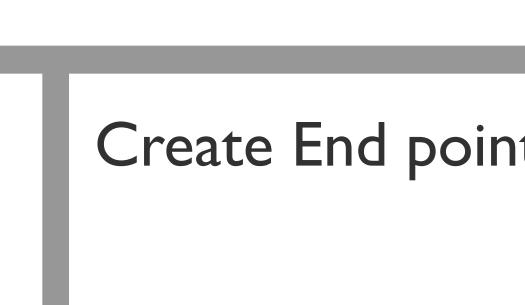
Create your schema here including your data types.

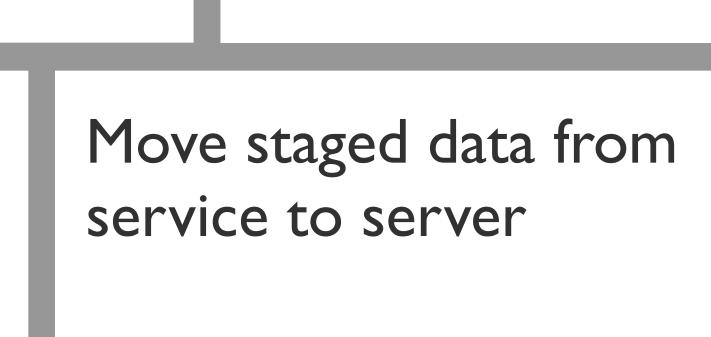
```
User:
 userName: {type: 'String', required: true},
 email: {type: 'String', required: true, unique: true},
  password: local auth & bcrypt,
  lastTeam: {ref to a team},
 loggedIn: {type: 'Boolean', required: true}
Chats:
  userId: {ref to user, required: true},
  timeStamp: {type: 'Date', required: true},
  teamRef: {ref to team, required: true},
  message: {type: 'String', required: true}
 teamName: {type: 'String', required: true, uniuqe: true},
 members: [{ref to members}],
  teamLead: {ref to member},
  regToken: {type: 'String', required: true},
  creationDate: {type: 'Date', required: true}
```

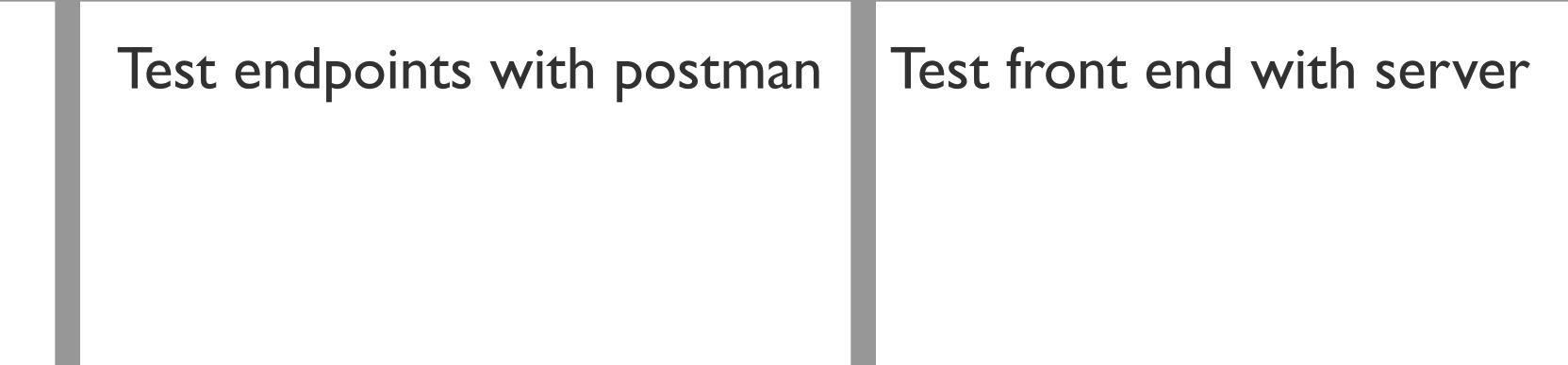
Views and Routes

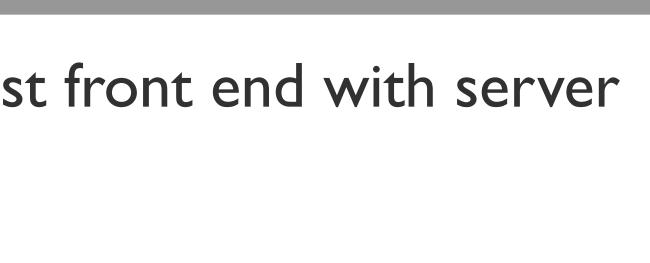
# staged data in service.

# Controllers and services with Test front end









Replace staged data with queries

Test with Postman

Test end to end

Get site hosted