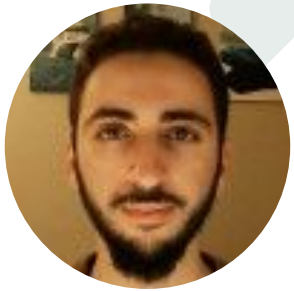
StudyTogether

Our Team

Daniel Laufer
Front-end Developer

Maor Gornic
Full Stack Developer

Milind Vishnoi
Front-end Developer



Mohamed Issa
Backend Developer

John Lewczuk
Full Stack Developer

Daniel Laufer

Front-end Developer

- Developed front-end features using React, Chakra+UI, and Redux, etc
- Integrated Google Maps into our app.
- Responsible for the front-end implementation of various core features of our app

chakra

Mohamed Issa

Back-end Developer

- **Backend-end**
  - MongoDB, NodeJS + express
- Notification system
- Authorization & Authentication on the backend

Maor Gornic

Full Stack Developer

- **Front-end**
  - React, Chakra UI
  - Study group view
- **Back-end**
  - MongoDB, NodeJS
  - Private study groups functionality
  - Invite users to study groups

John Lewczuk

Full Stack Developer

- **Front-end**
  - Email verification status
- **Back-end**
  - Email verification, recurring study groups

Milind Vishnoi

Front-end Developer

- Developed front-end features using React and Chakra+UI
- Created user page, study group components, following/follower page, etc.

chakra

Our App

Studying with others is an effective way of preparing for assessments

however it can be challenging to find others to study with at times.

*That's where **StudyTogether** comes to the rescue!*

Our App

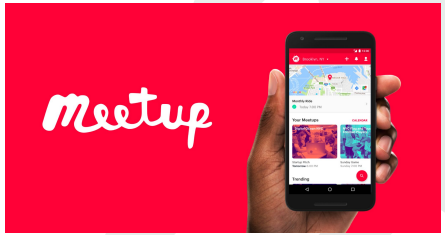
- StudyTogether facilitates the process of forming study groups at universities across Canada
- Students can easily form study groups that others can join
- Users can build connections through the app
  - Ability follow other users and get notified when they create a new group
  - Personalized profiles
  - Etc.
- A much, much more!

StudyTogether

Demo Time!

Our Competition

- Eventbrite
- Meetup
- Eventzilla
- Splash



What makes us unique?

- **Narrow scope**
  - Only used by students and tutors
  - Not clogged with irrelevant events
- **Student focused**
  - Allows us to create features just for students
  - Hosting office hours, etc.

Technologies Used

chakra

node JS

Firebase

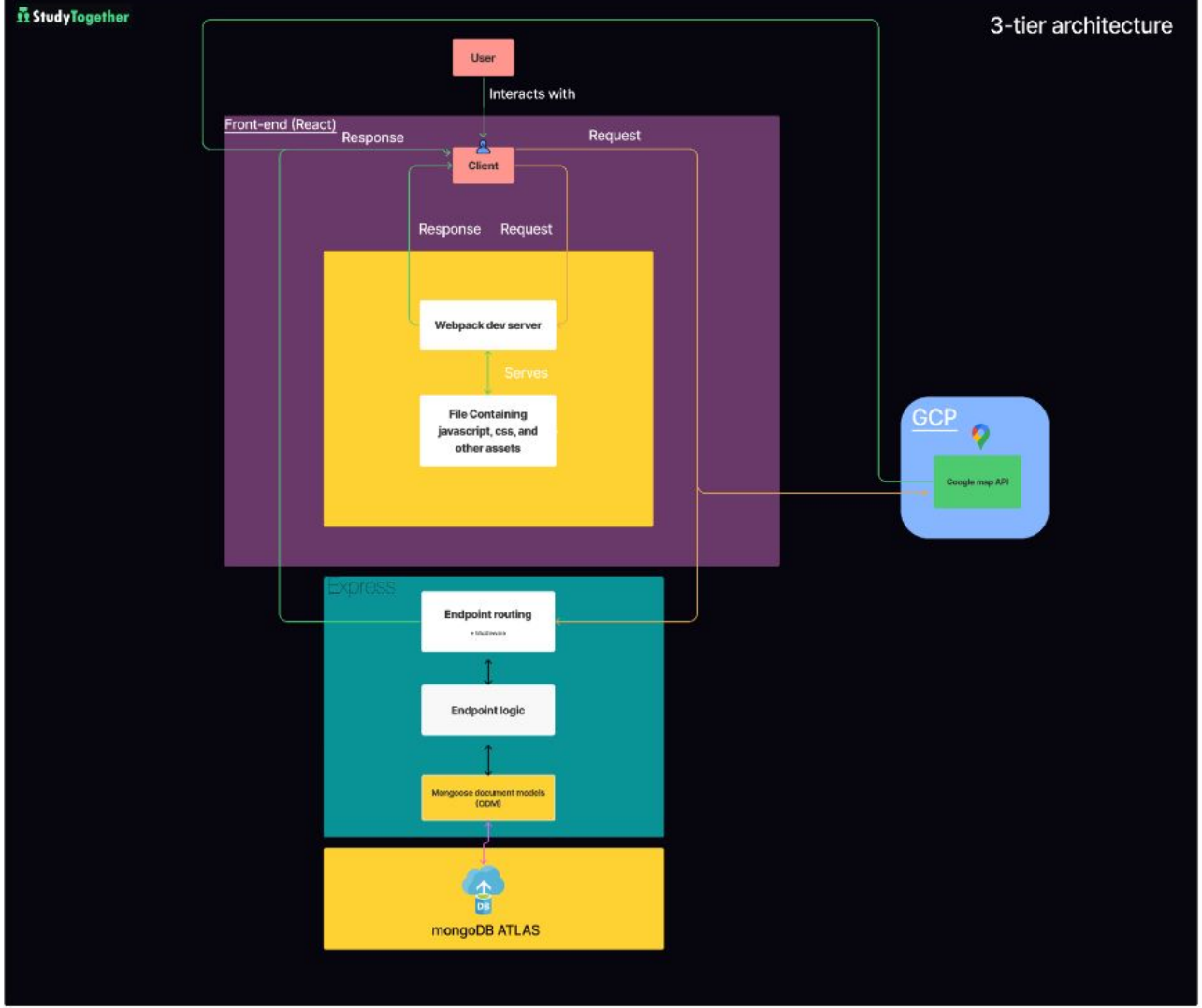
mongoDB

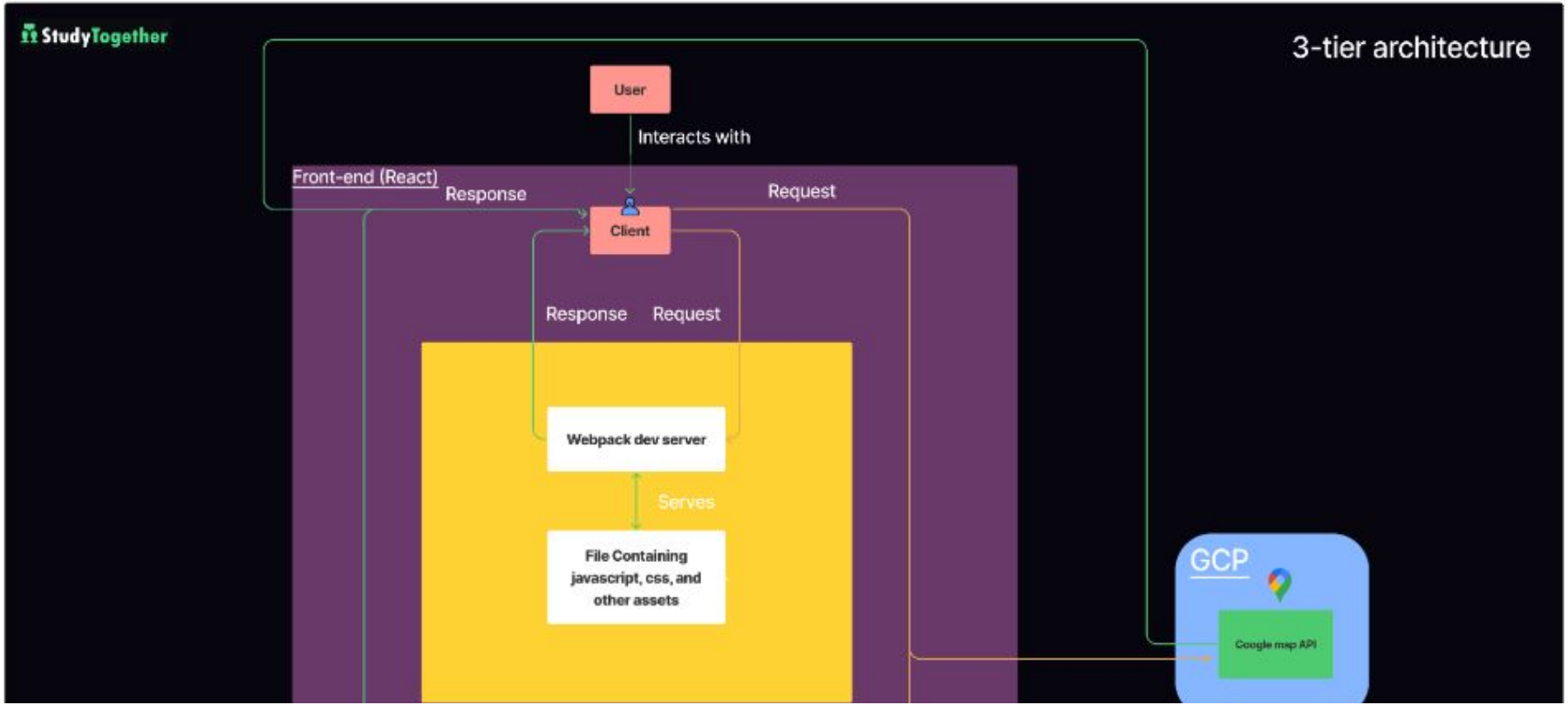
Express JS

Redux

Main Components

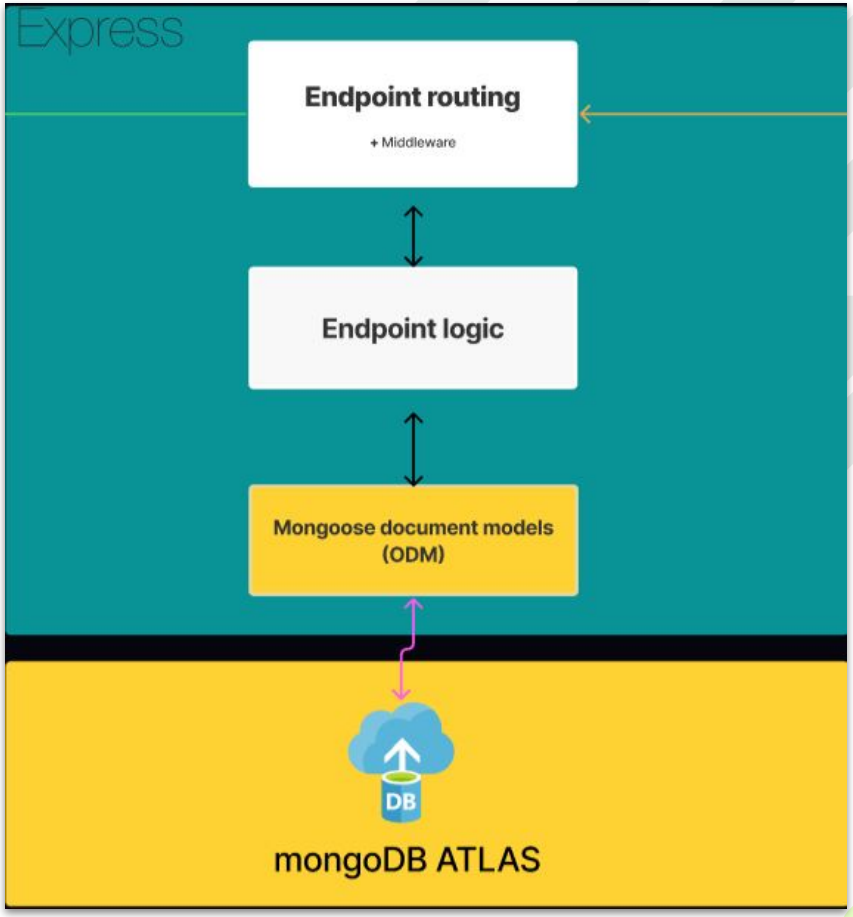
- React + Chakra UI
  - Used for our front-end development
- Redux
  - Used for state-management on the front-end
- ExpressJS + NodeJS
  - Used for our back-end
- MongoDB
  - Our database


StudyTogether
3-tier architecture
User
Interacts with
Front-end (React)
Response
Request
Client
Response
Request
Webpack dev server
Serves
File Containing javascript, css, and other assets
GCP
Google map API
Express
Endpoint routing
Endpoint logic
Mongoose document models (ODM)
mongoDB ATLAS


StudyTogether
3-tier architecture
User
Interacts with
Front-end (React)
Response
Request
Client
Response
Request
Webpack dev server
Serves
File Containing javascript, css, and other assets
GCP
Google map API

Application Tier

Data Tier



React, Chakra-UI, Redux

- Used React to create all our front-end components
- We used some pre-made Chakra-UI components
  - Ex. Loading spinners
- Redux was used for state management
  - Ex. Authentication was done using redux. (so we could easily pass authentication details to all the components that need it)

ExpressJS + NodeJS

- Used for our back-end development and creating our server
  - User requests are processed here
- Receives requests from the frontend, processes them, queries the database if needed and sends back a response.

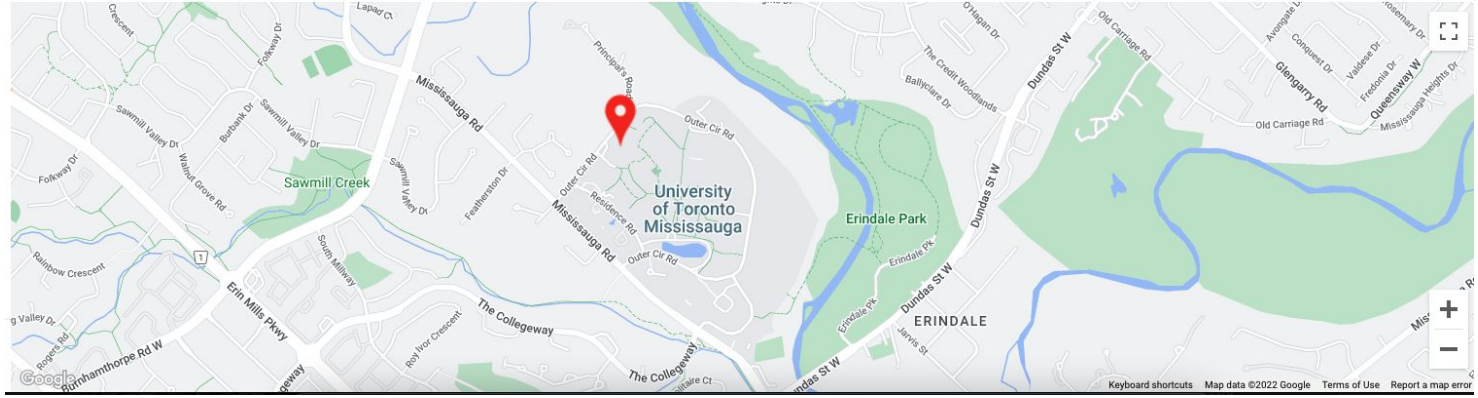
MongoDB

- Directly communicates with our back-end and ensures data becomes persistent
    - When a user creates a study group or makes changes to their profile, those changes will be saved
- Connected to the front-end
    - Upon a request to the server, the server serves this request accordingly and asks the database to store the given data.

mongoDB

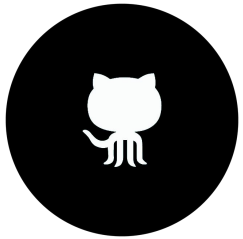
Software challenges

1. Google-maps-react
   - Integrating Google Maps into our app was a challenge.
   - Documentation wasn't great and ran into a lot of issues



Process

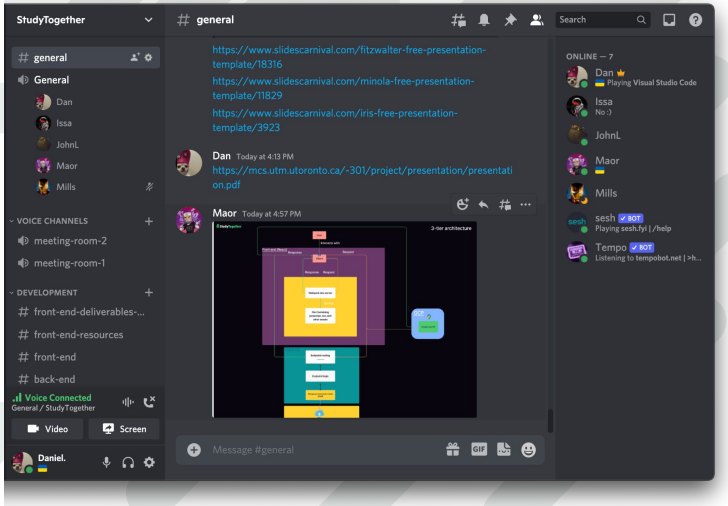





Discord

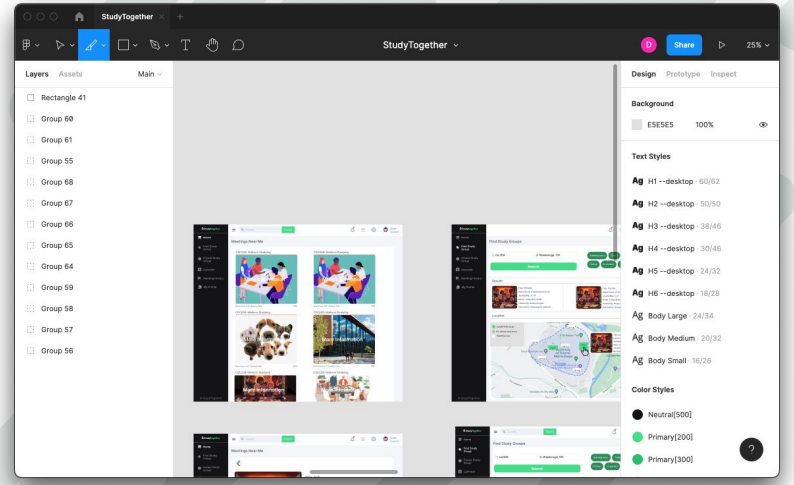
Main method of communication

- Updating other members on progress and deadlines
- Reaching out to other members for help
- Resolving conflicts in case there are any
- Requesting opinions on design choices
- Dedicated channels
  - Wiki, Frontend, Backend, reminders…



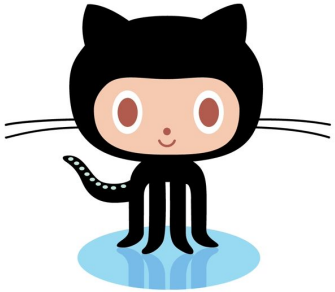
Figma

- Creating UI/UX design concepts.
- Help model components before creation.



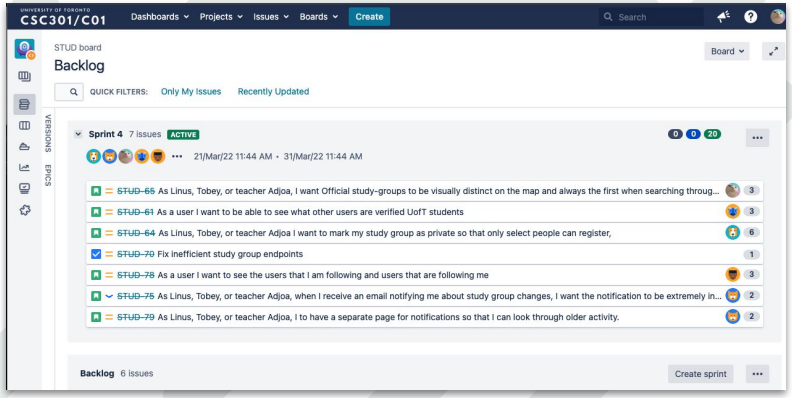
Github

- Our choice of version control
- Followed Git flow standard
- Requesting other member to review PR requests
- Provide technical and detailed comments on what can be improved



Jira

- Create and track tickets for each user story.
- Separated user stories into subtasks to help manage the workload.
- Add comments on tickets to clarify details for other members.
- Analytics: Burndown chart



Jira Software

Google Docs

Documentation and deliverables

- All derivables were collaboratively worked on through shared docs
- Keeping records of Bugs to fix.

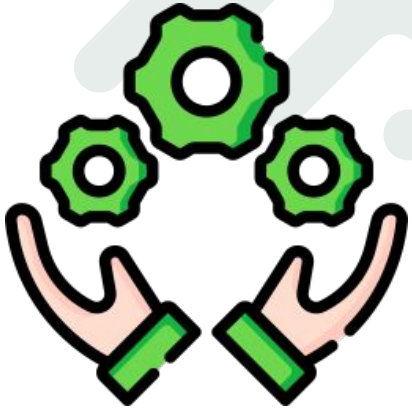
Practices

Bad practices as a team

- Resolving merging conflicts
  - Ex. Overwriting someone's work, introducing bugs, etc.
- Slow to review PRs
- Not updating Jira as soon as the PR is merged

Good practices as a team

- Active communication and fast replies on Discord
- Discussing issues as they arise
- Creating and editing documentation as project progresses



Documentation

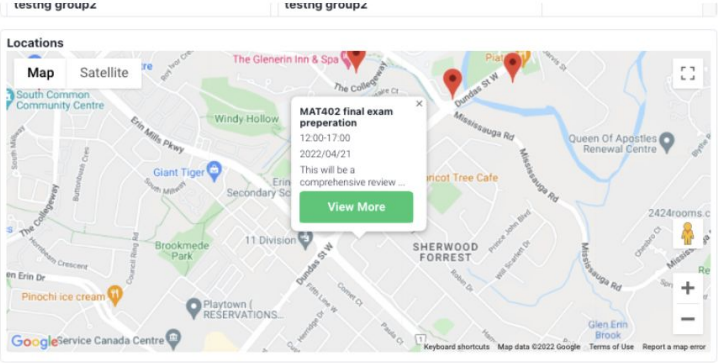
We divided the documentation into two sections:

- Frontend: Created documentation on google docs
- Backend: Utilized SwaggerUI and google docs

Documentation Standards - Frontend

- Mention the type of each props
- Attached screenshots for each component/page
- Created CRC cards for each component/page.

MarkerInfoWindow



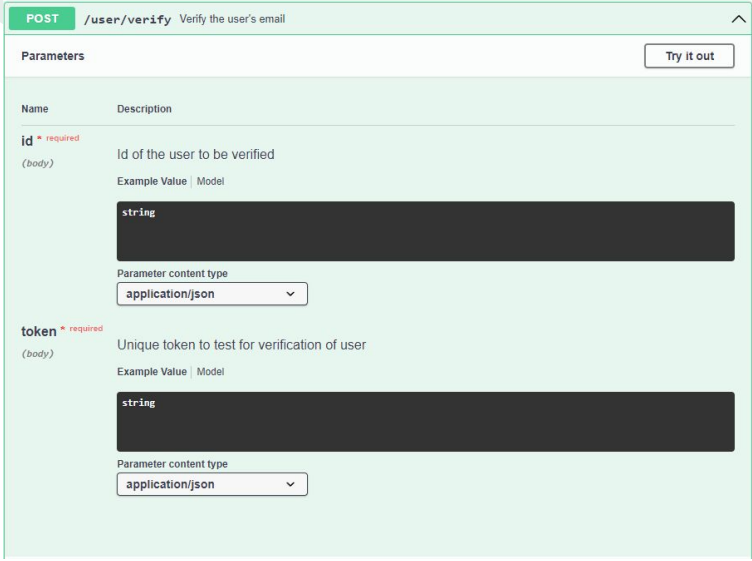

```
MarkerInfoWindow.propTypes = {
  group: PropTypes.PropTypes.shape({
      lat: PropTypes.number,
      lng: PropTypes.number,
      id: PropTypes.string,
      metaData: PropTypes.shape(),
    })
.isRequired,
};
MarkerInfoWindow.defaultProps = {};
```

Documentation Standards - Backend

- Created CRC cards for each file
- Swagger entry for each endpoint

Class Name: routes/forgot.js	
Parent Class: None **Subclasses:** None	

Responsibilities:
• Recover password in case user forgot it (i.e update old password to a new one)
  ○ Recover process involves tokenization to add a layer of security
  ○ Sends reset link to the user's email
	Collaborators:
	• models/user.model.js
	• models/token.model.js
	• app.js



Thanks!

- **Hope to see you on StudyTogether!**