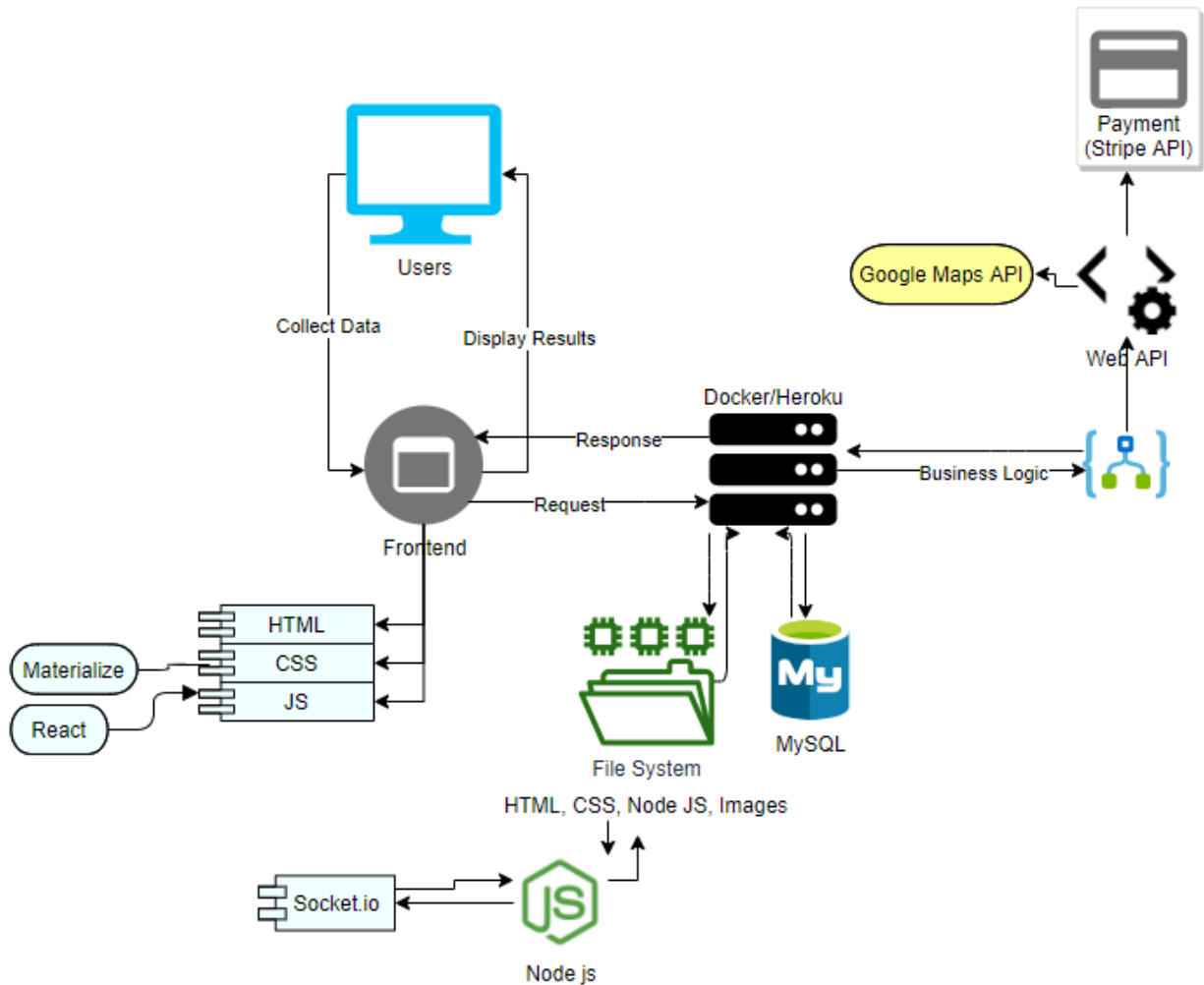


## Revised List of Features

- Feed (story points: 3)
  - Feed where the tasks are displayed from the database
  - As a user I would like to modify/filter my task area/feed
- Login/Register page (story points: 1)
  - Has to connect to the back end to check if they have an account and set new accounts
  - Use JWT cookie to store login
- Add tasks (story points: 1)
  - A page where the taskers can post the name of their tasks descriptions, name and price
- Complete task (story points: 4)
  - Marks task complete
  - Completed tasks should not show up in the Feed, but should be viewable on a user's history page
- Rating & feedback (story points: 5)
  - Place where customers can give other users a rating and review after they have completed a task from/for them. Will be stored in tasks table in mysql database
- Profile Page (story points: 1)
  - Display photo, name, bio, and rating (for each user this will be stored as a java script object and will be stored in the database)
  - List tasks this user has completed in the past (user rating can be calculated based off these)
  - Lists tasks a user has posted in the past
- Landing page (story points: 2)
  - This will have a description of the app and its functions as well as the links to register or login
  - Maybe display sample tasks/allow the user to browse the site to decide if they want to sign up
- Message board (story points: 3)
  - This is where the taskers and the workers will communicate logistics
- Notifications (story points: 3)
  - As a user I would like to be notified when my task is accepted or completed, or when I get a message.
- Payment (story points: 5)

## Architecture Diagram



### Front End design

- [https://github.com/CSCI-3308-CU-Boulder/3308SP21\\_011\\_6/tree/main/mileStones/wireframes](https://github.com/CSCI-3308-CU-Boulder/3308SP21_011_6/tree/main/mileStones/wireframes)
- <https://www.figma.com/file/ArJbsDpvzdTRAI4mkqG9mR/Taskit?node-id=0%3A1>

### Web Service Design

- Stripe API - A payment api
  - Stripe API has support for node.js
- Google Maps API - Allows developers to retrieve data from google maps
  - Goal of using Google Maps API is to perform a **radius search** in order to display tasks within a certain radius.

- Using the tasker/worker location we will iterate through our address table and filter tasks that are within a certain radius.

### Database design

Our database will be MySQL v8.0.23.

Database documentation has also been included here:

[https://github.com/CSCI-3308-CU-Boulder/3308SP21\\_011\\_6/tree/main/data](https://github.com/CSCI-3308-CU-Boulder/3308SP21_011_6/tree/main/data)

And will be updated if any structural changes need to be made during the development process.

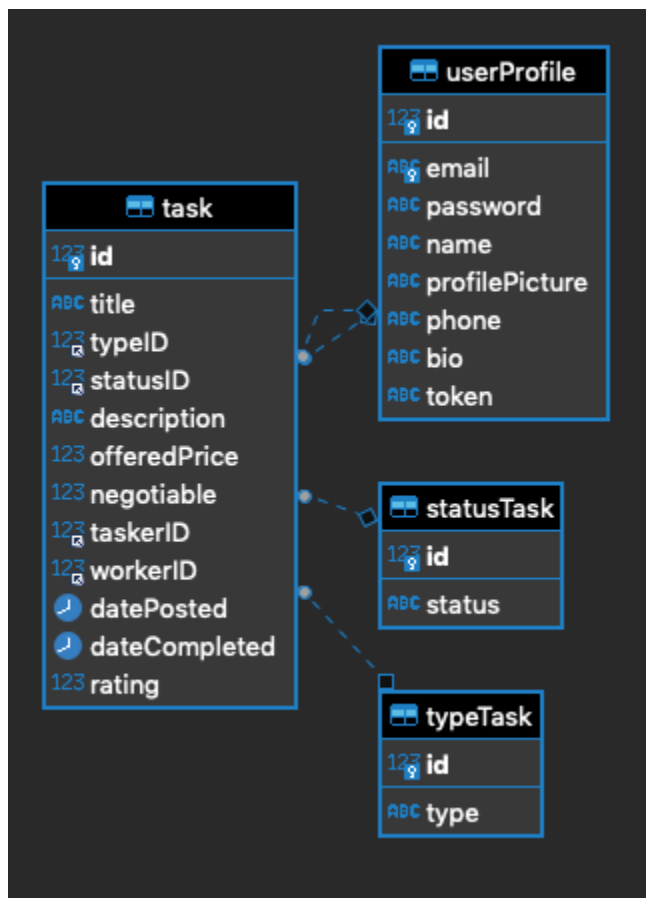


Diagram created with DBeaver Community

Edition

### task

column	type	properties	notes
id	int	PK,U,NN	
title	varchar64	NN	
typeID	int	NN	

statusID	int	NN	statusID defaults to 1 ('Pending') on creation, indicating the task is open for workers to accept.
description	longtext		
offeredPrice	float		
negotiable	tinyint	NN	1/0 indicates the price is/is not negotiable. This may or may not be implemented, so this currently defaults to 0.
taskerID	int	NN	
workerID	int		
datePosted	date		
dateCompleted	date		
rating	int		

#### **typeTask**

<i>column</i>	<i>type</i>	<i>properties</i>	<i>notes</i>
id	int	PK, U, NN	
type	varchar128		

#### **statusTask**

<i>column</i>	<i>type</i>	<i>properties</i>	<i>notes</i>
id	int	PK, U, NN	
status	varchar128		

#### **userProfile**

<i>column</i>	<i>type</i>	<i>properties</i>	<i>notes</i>
id	int	PK, U, NN	
email	varchar128	NN, U	
password	varchar128	NN	This should be a hash of the password, not the actual pw itself
name	varchar128	NN	
profilePicture	varchar128		Link to picture resource (actual image stored in S3 bucket, or heroku equivalent?)
phone	varchar20		
bio	longtext		
token	varchar128		Jwt refresh token for use with login

PK = Primary Key (these auto increment used to index the table)

NN = Non Nullable (these are required fields)

U = Unique (may not insert the same value more than once)

### Individual Contributions

Timothy: working on the feed and connecting it with the backend to iterate through the tasks and display them

Jules: Has been working on View Profile and Edit Profile pages. Will connect with the database for get and post calls.

Manoj: Have been working on connecting frontend registration page (React) to backend (node.js)

[https://github.com/CSCI-3308-CU-Boulder/3308SP21\\_011\\_6/commit/6bf1f1232a09065661a232ad356e2b8f7929053a](https://github.com/CSCI-3308-CU-Boulder/3308SP21_011_6/commit/6bf1f1232a09065661a232ad356e2b8f7929053a)

Josh: Working to connect frontend to backend with fetch later using axios, learned to use insomnia, learning to connect to backend with mysql using workbench, update architecture diagram, researched on web api services, tasked to work on chat service using node.js, react.js, and socket.io

Greg: Working on Login functionality, helped debug/resolve mysql errors

[https://github.com/CSCI-3308-CU-Boulder/3308SP21\\_011\\_6/commit/c756d18716c1718d5d2eb4954e9988ee789edf82](https://github.com/CSCI-3308-CU-Boulder/3308SP21_011_6/commit/c756d18716c1718d5d2eb4954e9988ee789edf82)

Ryan: Landing page interface and filter tasks page(using React and node.js). Going back to add modal and still need to run code on localhost:3000 server. Filter tasks page may not be needed, instead a personal preference page w/ sorting algorithm.

Epic		
✚ Y12-6 Account Management		
▼ ✚ Y12-17 DevOps (Stuff for developers, not customers)		
📌 Y12-18 As a team, we need a dockerfile so we can work in identical environments	TO DO	👤
📌 <del>Y12-19</del> As a developer I need a database to persist data	COMPLETE	👤
▼ ✚ Y12-33 Epic 1: Registration and Authorization		
📌 Y12-2 As a new user I want to create new account,so I can access the application	TO TEST	MY
📌 Y12-7 As a user, I need a login page to keep my account secure	IN PROGRESS	👤
✅ Y12-67 Plug Back End into Front End - Registration	IN PROGRESS	MY
▼ ✚ Y12-34 Epic 2: User Account		
📌 Y12-16 As a tasker, I want to see highly rated workers in my feed so I can be sure that the work I assign to them will be don...	TO DO	JF
📌 Y12-8 As a user, I would like to edit my profile so I can keep my info up to date.	IN PROGRESS	JF
✅ Y12-68 Plug Backend into Frontend	TO DO	
▼ ✚ Y12-35 Epic 3: Website		
📌 Y12-29 As a user, I want a landing page to determine whether I want to sign up for Taskit services.	TO TEST	RR
📌 Y12-3 As a laborer, I want to see a list of tasks so I can select a job	IN PROGRESS	TW
📌 Y12-25 As a user I need a Feed where all tasks can be displayed	IN PROGRESS	TW
📌 Y12-48 As a user, I want a discovery page so I can consider whether I want to hire more workers for a project.	TO DO	
📌 <del>Y12-65</del> As a user, I need a workspace page where I can see my own tasks	COMPLETE	JT
▼ ✚ Y12-1 Epic 4: Job Posting Board		
📌 Y12-43 As a user, I want a page where workers can create their own tasks. And add in descriptions, levels of pricing...	IN PROGRESS	JT
📌 Y12-12 As a user, I would like to modify/filter my task area	IN PROGRESS	RR
📌 <del>Y12-9</del> As a user, I need a the company to have a name	COMPLETE	JF
✅ Y12-69 Plug Backend into Frontend	IN PROGRESS	JF
▼ ✚ Y12-37 Epic 5: Message/Notification Board		
📌 Y12-5 As a worker, I would like a message/notification board to communicate with the people when my task is accepted or...	TO DO	TW
📌 Y12-45 As a user, I need a message board to communicate with workers	TO DO	RR
📌 Y12-49 As a worker/tasker, I would like to send files through your messaging app. If not then I'll use my email.	TO DO	JT

## Challenges

- 1.) Figuring out how to connect react to node.js had to read different sources to find this information
- 2.) Connecting to mysql workbench (error in scripts) - Resolved as of March 17
- 3.) Need to make modifications to database structure to accommodate certain features now that back end and front end are being connected.
- 4.) Not much thought has gone into security yet

Identify at least 3 challenges and/or risks to your project at this point.

What is your backup/risk mitigation plan incase you are unable to resolve the challenge(s)?

Solutions

1. Using Axios to make http requests from React
2. Found an error in scripts and corrected it.
3. Continue to adapt back-end as the needed structure becomes more apparent
  - a. Will require team members to communicate on changes that are needed
  - b. Need to make sure edits don't break previously written code