Meas / itroadway-test

0 stars 1 fork

Star Watch

Code Issues Pull requests Actions Projects Wiki Security

main ...

Meas initial draft 2 of the test specs ... on Dec 30, 2020 5

View code

README.md

ITRoadway-test

Introduction

Hello there potential ITRoadway developers. The App that you will be building will be a very simple ToDo application, with a focus on seeing how you handle technologies you probably never used before.

We would also like to see how you work together in a team, so while there will be distinct tasks for both frontend and backend, some tasks will be shared ones where you can coordinate between yourselves about who will do which part.

This whole test is of an open ended nature, so in case some things are not clearly defined, it will be up to you to make the decision and explain your choice about going that route on the interview. Of course, you can still feel free to contact us about any potential clarifications about the tasks listed.

Summary

The idea of the app that a user can register/login to the App and is able to create new notes and to update/delete previous ToDos. ToDos themselves are simple entity which has a title, description and timestamps.

Shared

- Setup a Laravel application in such a way that the Vue.js part will act as a SPA (single page application) and the Laravel part as a RESTful API
- Users should only be able to perform CRUD operations only on their own ToDos

Frontend

- There will be 6 screens in total: welcome screen (for guest users), login, register, ToDos list, create new ToDo, view/update ToDo
- Once the user is logged in, they should stay logged in (even if the page is refreshed) using the existing token
- Logout button in navigation which will logout the user
- There are no designs provided, and you will have to make the screens as you see fit
- **Welcome screen**: will only be visible if a user is not logged in and is on route '/'
- **Register screen** :
  only accessible to guest users (not logged in)
  only 2 fields are required - namely email and password
  upon successful registration; the user will be notified to check their email inbox and redirected to the login screen
- **Login screen** :
  only accessible to guest users
  2 fields - email and password
  if credentials are wrong the user should be notified
  if credentials are correct the user should be logged in and redirected to '/' (which is now the list of my ToDos screen)
- **ToDo list** :
  only accessible for logged in users
  the user should see a list of their previously created ToDos
  the user should be able to go to the "Create new ToDo" screen from this one
  the user should be able to go to the "View/update ToDo" screen by selecting one of

the ToDos from the list
the user should be able to delete a particular ToDo from the list
ToDos should be searchable by title
Pagination should be handled in an "infinite scroll" kind of way

- **Create ToDo screen** :
  only accessible for logged in users
  2 fields: title and description of the ToDo
  upon successfully creating the ToDo the user should be redirected to the "ToDo list" screen
- **View/update ToDo screen** :
  only accessible for logged in users
  2 fields: title and description of the ToDo, and these fields should be pre filled with data from the server about this particular ToDo
  upon successfully updating the ToDo the user should be redirected to the "ToDo list" screen

Backend

- Creating the exact models needed for the application to work properly is completely up to you
- Database used should be MySQL
- **Register route** :
  required email and password
  route should create a user that is not verified in the database
  route should send a verification email to the users email
- **Verification route** :
  if the verification code is found in the database, the user should now be verified
- **Login route** :
  check if email and password are correct and that the user is verified, otherwise reject
  if user is verified and email/password are correct, respond with a JWT token
- **Logout route** :
  invalidate the user's JWT token
- **ToDo list route** :
  only accessible for logged in users
  send out a paginated list of ToDos (up to 10 items per page, preferably use Laravel's built in pagination)

the list should be searchable by name (work together with the frontend to figure out the best way to get the search text)

- **Create ToDo route** :
  only accessible for logged in users
  2 fields required: title and description of the ToDo
  on success save the item to the database
- **View ToDo route** :
  only accessible for logged in users
  respond with an object containing the title and description of the requested ToDo
- **Update ToDo route** :
  only accessible for logged in users
  2 fields required: title and description of the ToDo
  on success save updated item to the database
- **Delete ToDo route** :
  only accessible for logged in users
  on success delete the item from the database

Closing tips

- Since the app is simple by nature, the focus will be on covering all use cases, sending out appropriate responses / notifying the user properly, and clean code

Releases

No releases published

Packages

No packages published