AppointmentNow

Alex, Jackson and Gavin

AppointmentNow

Function

AppointmentNow aims to solve the problem of small businesses needing a modular appointment booking system that can be customised to their needs without having to commision expensive, bespoke software. It aims to have an inutiative interface with granular complexity.

Roles

Roles

Alex

- Project Manager
- Front End
- REACT
- Rails/REACT integration
- Rails Controllers

Roles

Jackson

- Primary Rails Logic
- Backend
- Rails/REACT integration

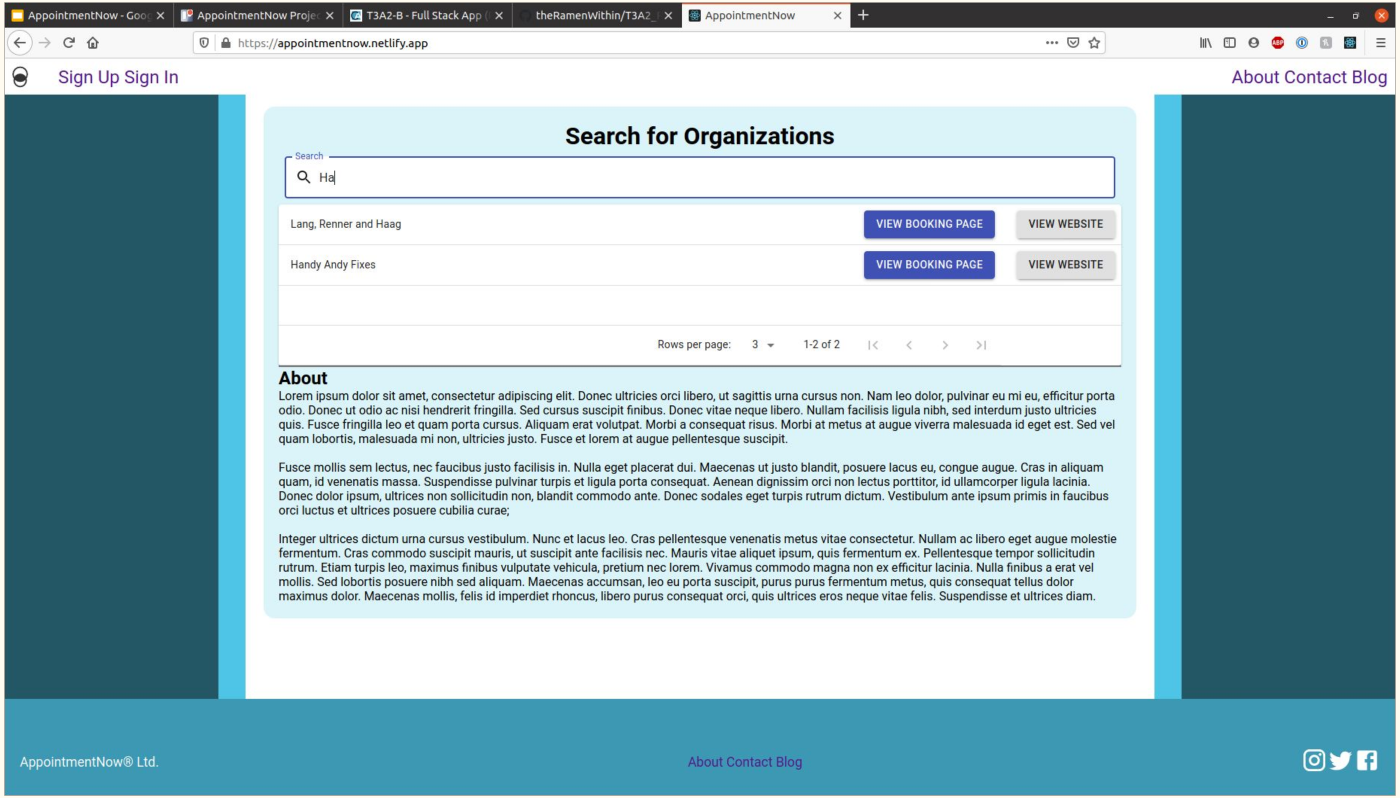
Roles

Gavin

- Testing
- Hosting
- Auxiliary Rails Logic
- Backend Hosting / Heroku

Front-End

Final Product



Search for Organizations

Search

Ha

Lang, Renner and Haag	VIEW BOOKING PAGE	VIEW WEBSITE
Handy Andy Fixes	VIEW BOOKING PAGE	VIEW WEBSITE

Rows per page: 3 1-2 of 2

About

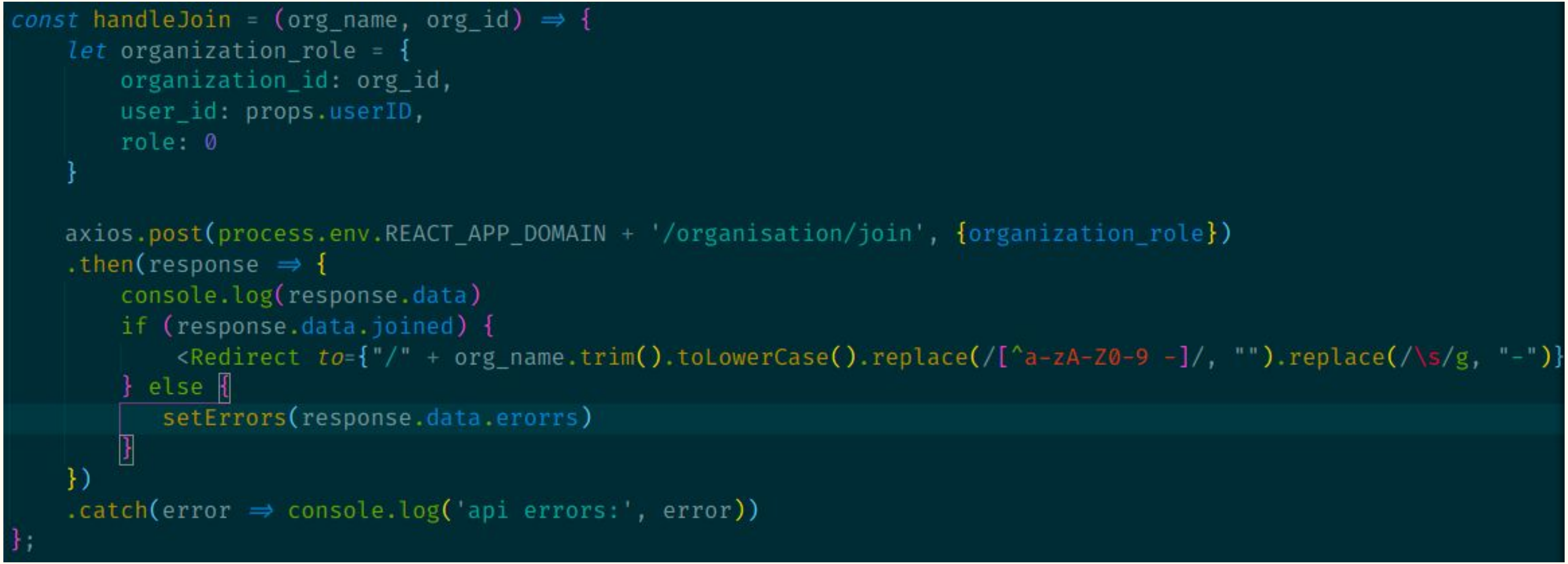
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ultricies orci libero, ut sagittis urna cursus non. Nam leo dolor, pulvinar eu mi eu, efficitur porta odio. Donec ut odio ac nisi hendrerit fringilla. Sed cursus suscipit finibus. Donec vitae neque libero. Nullam facilisis ligula nibh, sed interdum justo ultricies quis. Fusce fringilla leo et quam porta cursus. Aliquam erat volutpat. Morbi a consequat risus. Morbi at metus at augue viverra malesuada id eget est. Sed vel quam lobortis, malesuada mi non, ultricies justo. Fusce et lorem at augue pellentesque suscipit.

Fusce mollis sem lectus, nec faucibus justo facilisis in. Nulla eget placerat dui. Maecenas ut justo blandit, posuere lacus eu, congue augue. Cras in aliquam quam, id venenatis massa. Suspendisse pulvinar turpis et ligula porta consequat. Aenean dignissim orci non lectus porttitor, id ullamcorper ligula lacinia. Donec dolor ipsum, ultrices non sollicitudin non, blandit commodo ante. Donec sodales eget turpis rutrum dictum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae;

Integer ultrices dictum urna cursus vestibulum. Nunc et lacus leo. Cras pellentesque venenatis metus vitae consectetur. Nullam ac libero eget augue molestie fermentum. Cras commodo suscipit mauris, ut suscipit ante facilisis nec. Mauris vitae aliquet ipsum, quis fermentum ex. Pellentesque tempor sollicitudin rutrum. Etiam turpis leo, maximus finibus vulputate vehicula, pretium nec lorem. Vivamus commodo magna non ex efficitur lacinia. Nulla finibus a erat vel mollis. Sed lobortis posuere nibh sed aliquam. Maecenas accumsan, leo eu porta suscipit, purus purus fermentum metus, quis consequat tellus dolor maximus dolor. Maecenas mollis, felis id imperdiet rhoncus, libero purus consequat orci, quis ultrices eros neque vitae felis. Suspendisse et ultrices diam.

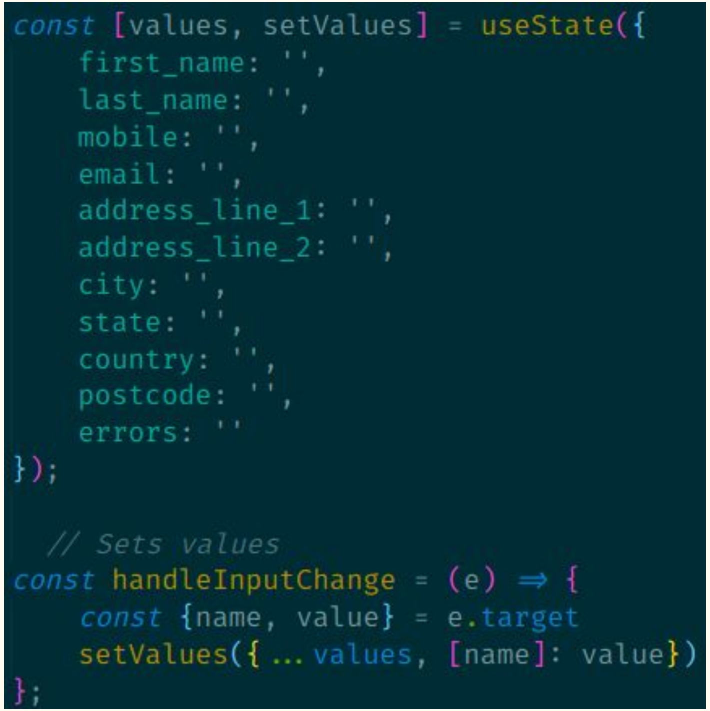
AppointmentNow® Ltd.

Axios

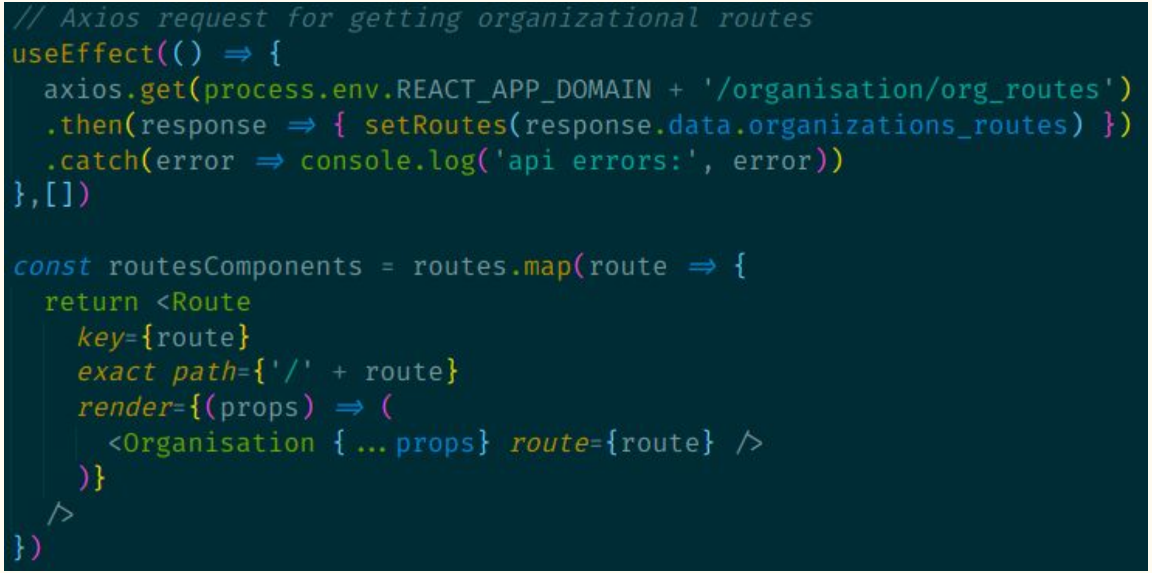


```
const handleJoin = (org_name, org_id) => {
    let organization_role = {
        organization_id: org_id,
        user_id: props.userID,
        role: 0
    }

    axios.post(process.env.REACT_APP_DOMAIN + '/organisation/join', {organization_role})
    .then(response => {
        console.log(response.data)
        if (response.data.joined) {
            <Redirect to={"/" + org_name.trim().toLowerCase().replace(/[^a-zA-Z0-9 -]/, "").replace(/\s/g, "-")}
        } else {
            setErrors(response.data.erorrs)
        }
    })
    .catch(error => console.log('api errors:', error))
};
```

Import React, {useState, useEffect} from ‘react’;

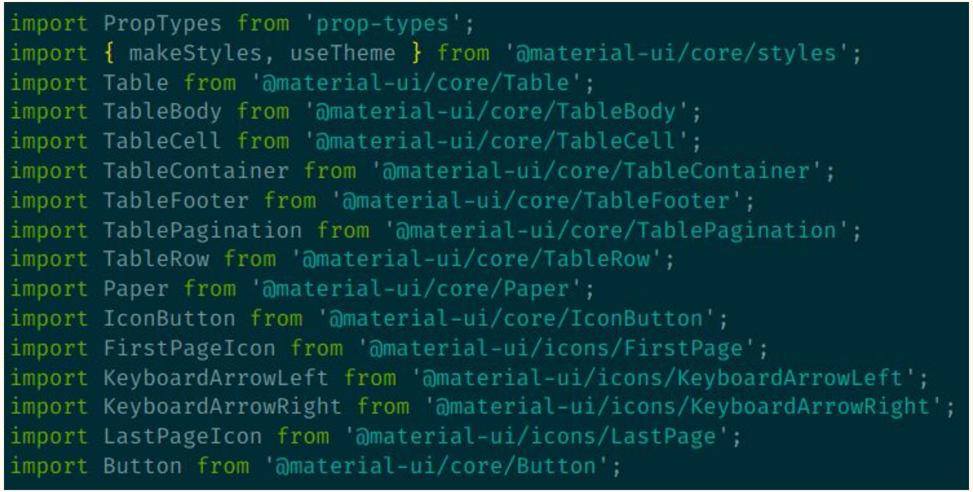


```
const [values, setValues] = useState({
    first_name: '',
    last_name: '',
    mobile: '',
    email: '',
    address_line_1: '',
    address_line_2: '',
    city: '',
    state: '',
    country: '',
    postcode: '',
    errors: ''
});

  // Sets values
const handleInputChange = (e) => {
    const {name, value} = e.target
    setValues({...values, [name]: value})
};
```

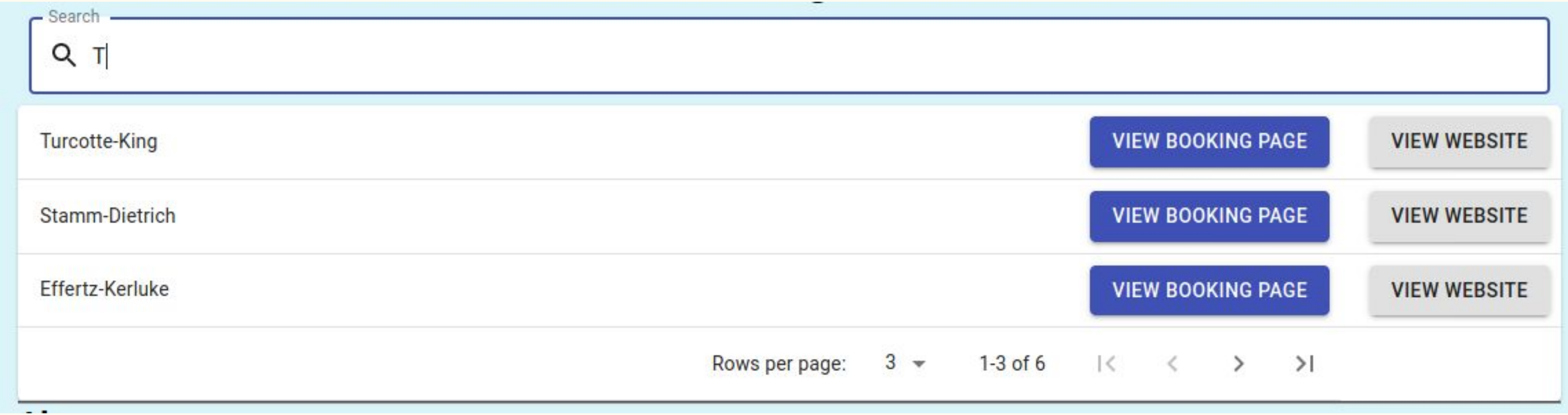


```
// Axios request for getting organizational routes
useEffect(() => {
  axios.get(process.env.REACT_APP_DOMAIN + '/organisation/org_routes')
  .then(response => { setRoutes(response.data.organizations_routes) })
  .catch(error => console.log('api errors:', error))
},[])

const routesComponents = routes.map(route => {
  return <Route
    key={route}
    exact path={'/' + route}
    render={(props) => (
      <Organisation {...props} route={route} />
    )}
  />
})
```

Material UI



```
import PropTypes from 'prop-types';
import { makeStyles, useTheme } from '@material-ui/core/styles';
import Table from '@material-ui/core/Table';
import TableBody from '@material-ui/core/TableBody';
import TableCell from '@material-ui/core/TableCell';
import TableContainer from '@material-ui/core/TableContainer';
import TableFooter from '@material-ui/core/TableFooter';
import TablePagination from '@material-ui/core/TablePagination';
import TableRow from '@material-ui/core/TableRow';
import Paper from '@material-ui/core/Paper';
import IconButton from '@material-ui/core/IconButton';
import FirstPageIcon from '@material-ui/icons/FirstPage';
import KeyboardArrowLeft from '@material-ui/icons/KeyboardArrowLeft';
import KeyboardArrowRight from '@material-ui/icons/KeyboardArrowRight';
import LastPageIcon from '@material-ui/icons/LastPage';
import Button from '@material-ui/core/Button';
```

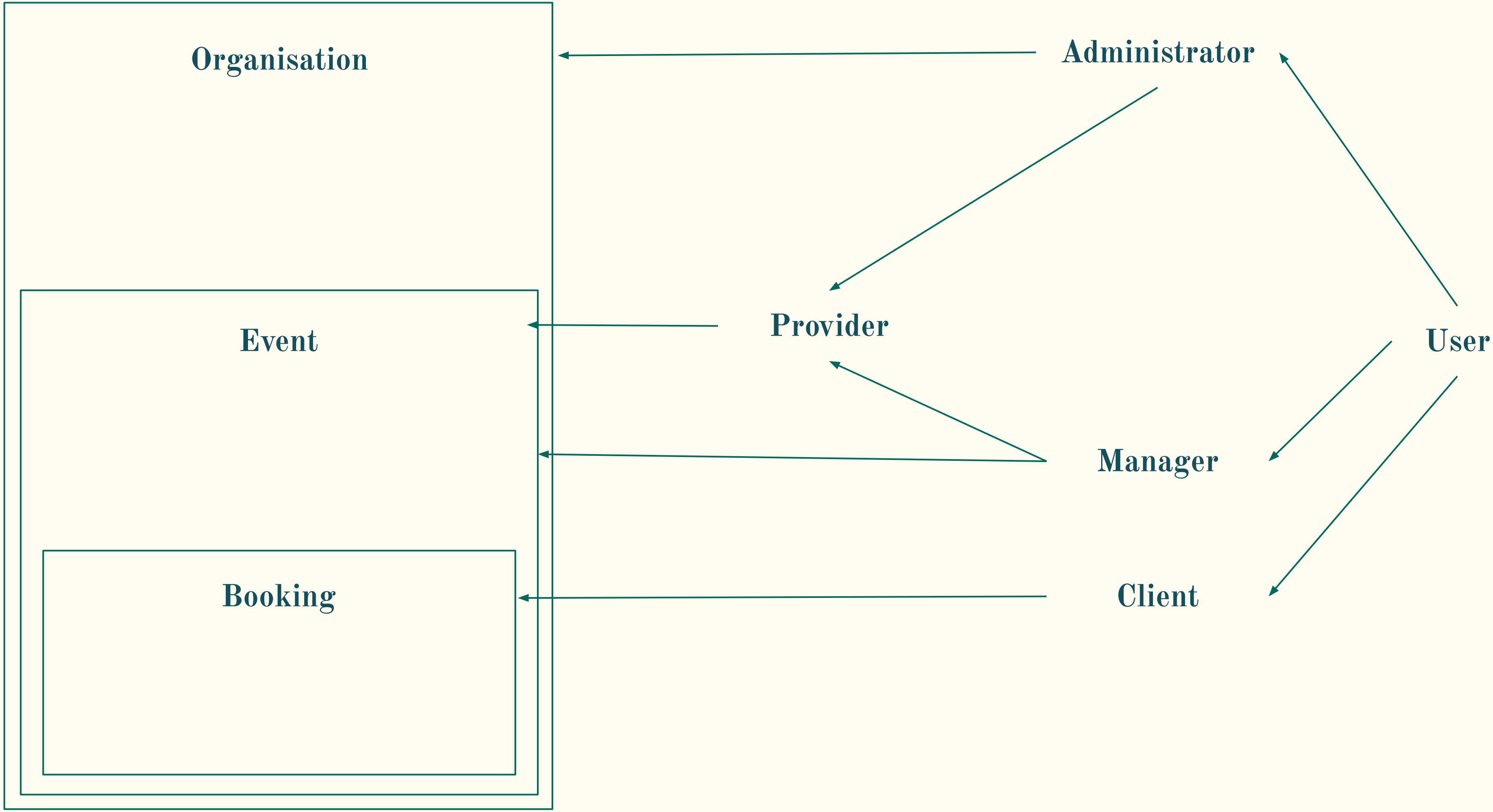


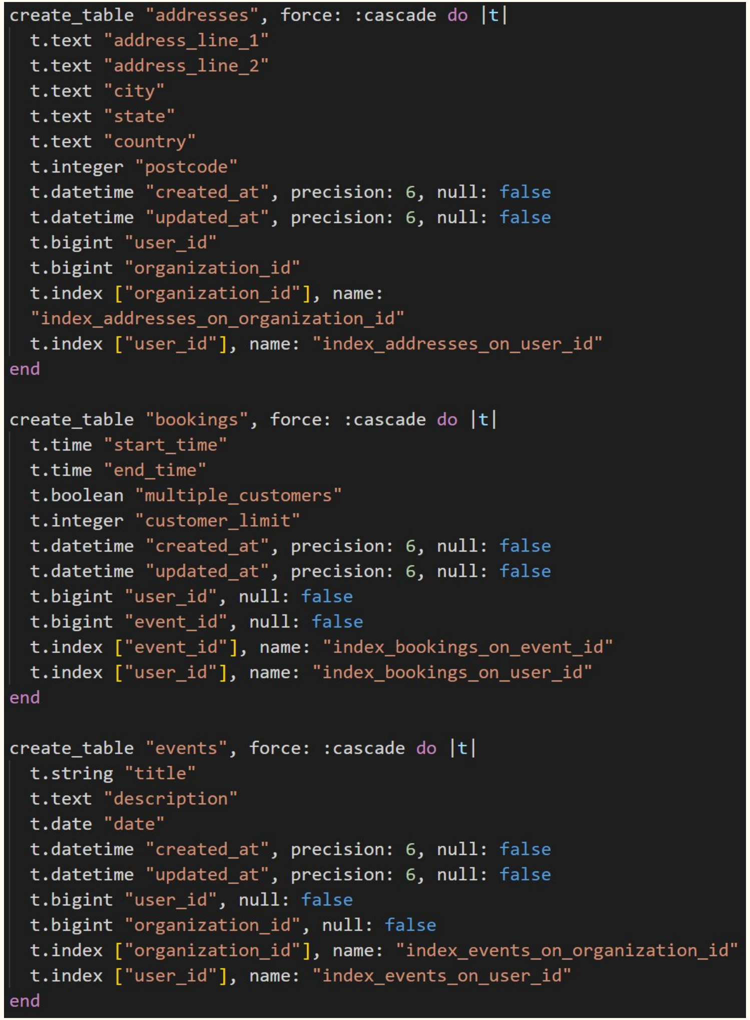
Search: T

Turcotte-King	VIEW BOOKING PAGE	VIEW WEBSITE
Stamm-Dietrich	VIEW BOOKING PAGE	VIEW WEBSITE
Effertz-Kerluke	VIEW BOOKING PAGE	VIEW WEBSITE

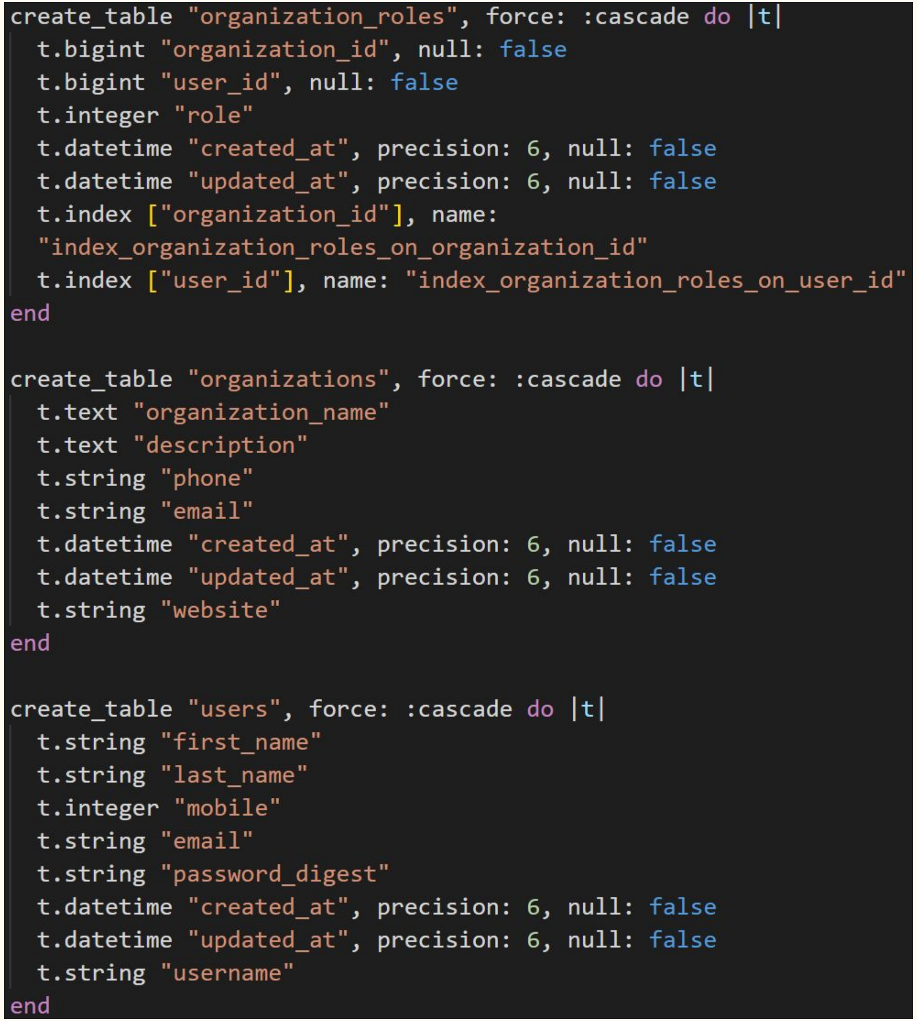
Rows per page: 3 1-3 of 6

Back-End

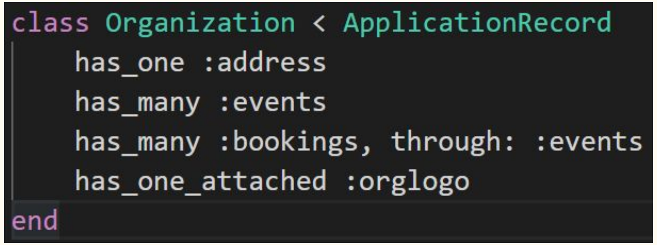

Organisation
Administrator
Provider
User
Event
Manager
Booking
Client



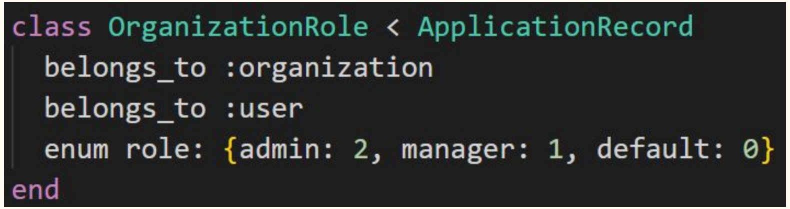
```ruby
create_table "addresses", force: :cascade do |t|
  t.text "address_line_1"
  t.text "address_line_2"
  t.text "city"
  t.text "state"
  t.text "country"
  t.integer "postcode"
  t.datetime "created_at", precision: 6, null: false
  t.datetime "updated_at", precision: 6, null: false
  t.bigint "user_id"
  t.bigint "organization_id"
  t.index ["organization_id"], name:
  "index_addresses_on_organization_id"
  t.index ["user_id"], name: "index_addresses_on_user_id"
end

create_table "bookings", force: :cascade do |t|
  t.time "start_time"
  t.time "end_time"
  t.boolean "multiple_customers"
  t.integer "customer_limit"
  t.datetime "created_at", precision: 6, null: false
  t.datetime "updated_at", precision: 6, null: false
  t.bigint "user_id", null: false
  t.bigint "event_id", null: false
  t.index ["event_id"], name: "index_bookings_on_event_id"
  t.index ["user_id"], name: "index_bookings_on_user_id"
end

create_table "events", force: :cascade do |t|
  t.string "title"
  t.text "description"
  t.date "date"
  t.datetime "created_at", precision: 6, null: false
  t.datetime "updated_at", precision: 6, null: false
  t.bigint "user_id", null: false
  t.bigint "organization_id", null: false
  t.index ["organization_id"], name: "index_events_on_organization_id"
  t.index ["user_id"], name: "index_events_on_user_id"
end
```



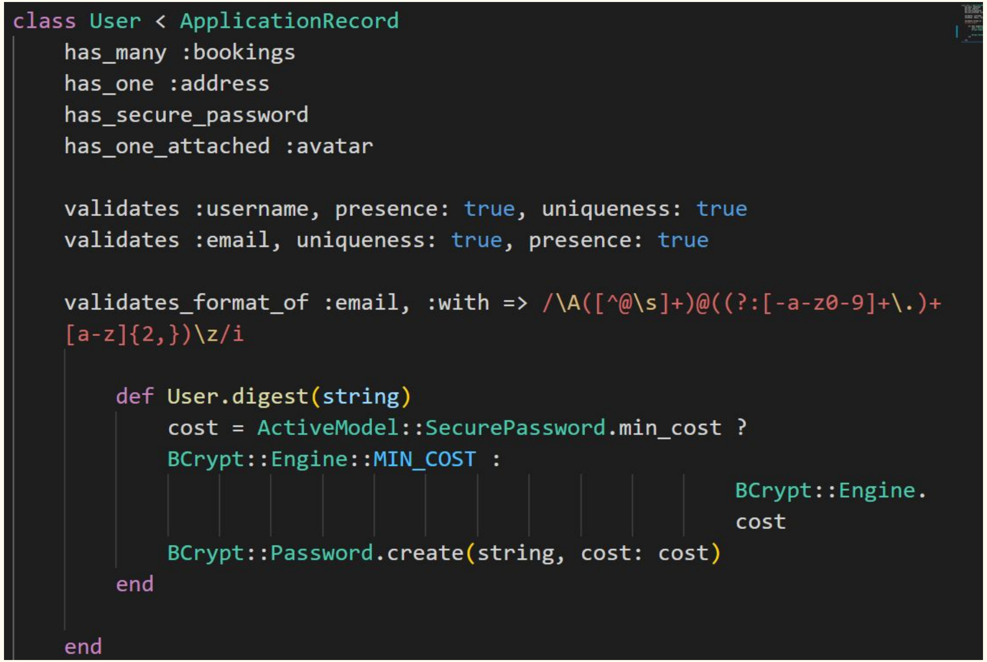
```ruby
create_table "organization_roles", force: :cascade do |t|
  t.bigint "organization_id", null: false
  t.bigint "user_id", null: false
  t.integer "role"
  t.datetime "created_at", precision: 6, null: false
  t.datetime "updated_at", precision: 6, null: false
  t.index ["organization_id"], name:
  "index_organization_roles_on_organization_id"
  t.index ["user_id"], name: "index_organization_roles_on_user_id"
end

create_table "organizations", force: :cascade do |t|
  t.text "organization_name"
  t.text "description"
  t.string "phone"
  t.string "email"
  t.datetime "created_at", precision: 6, null: false
  t.datetime "updated_at", precision: 6, null: false
  t.string "website"
end

create_table "users", force: :cascade do |t|
  t.string "first_name"
  t.string "last_name"
  t.integer "mobile"
  t.string "email"
  t.string "password_digest"
  t.datetime "created_at", precision: 6, null: false
  t.datetime "updated_at", precision: 6, null: false
  t.string "username"
end
```



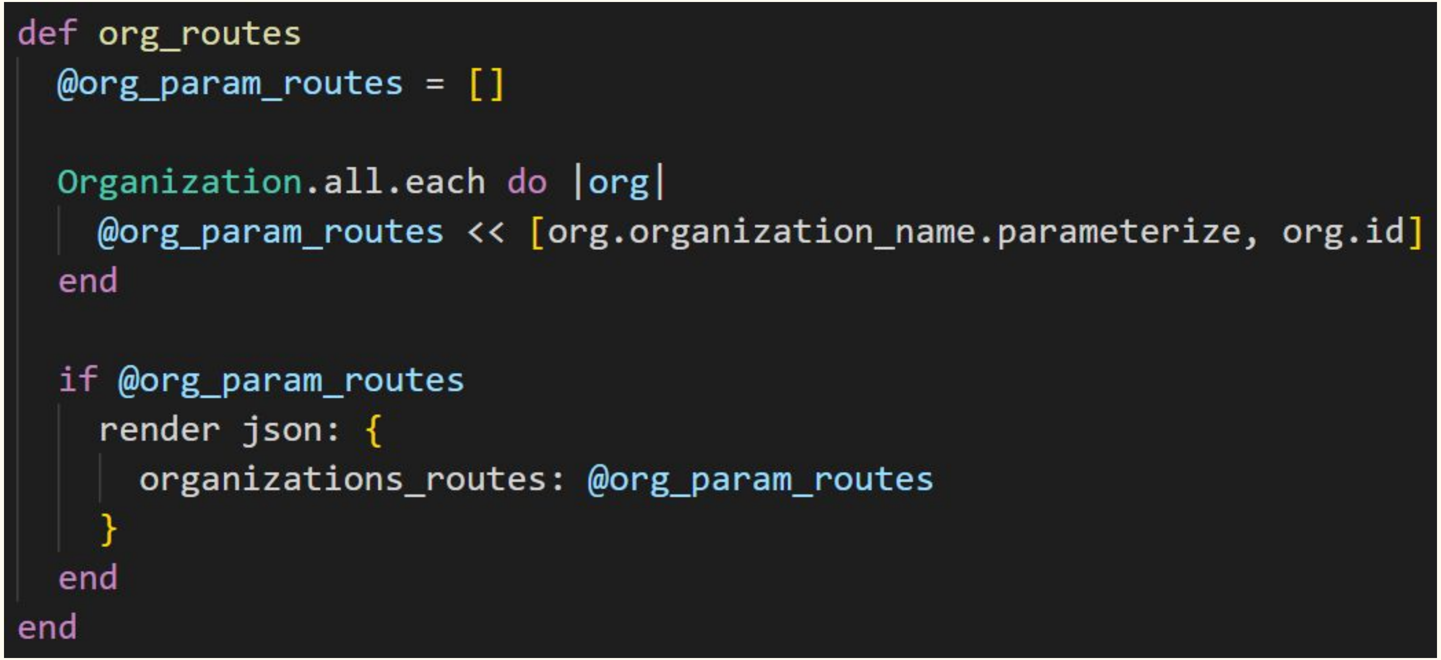
```ruby
class Organization < ApplicationRecord
    has_one :address
    has_many :events
    has_many :bookings, through: :events
    has_one_attached :orglogo
end
```



```ruby
class OrganizationRole < ApplicationRecord
  belongs_to :organization
  belongs_to :user
  enum role: {admin: 2, manager: 1, default: 0}
end
```

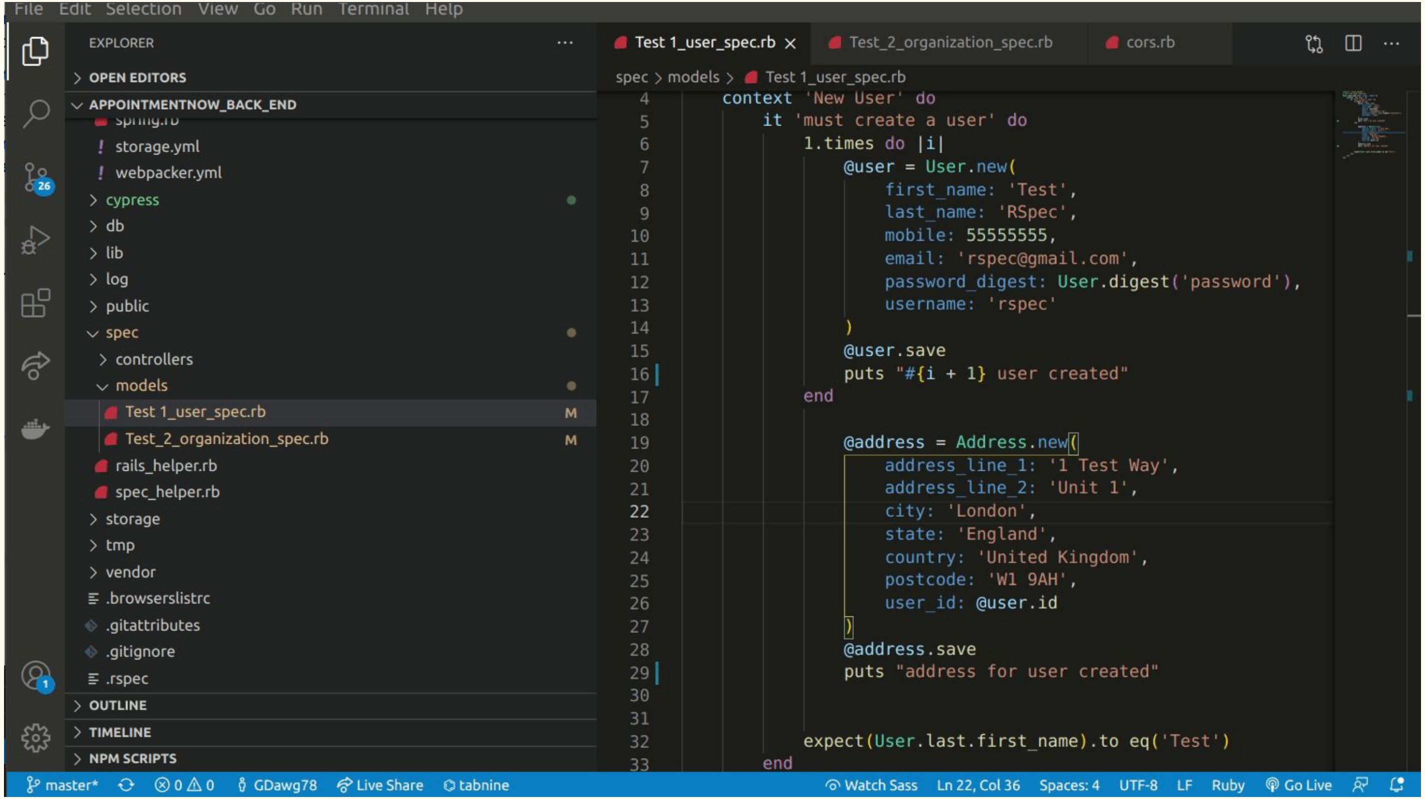


```ruby
class User < ApplicationRecord
    has_many :bookings
    has_one :address
    has_secure_password
    has_one_attached :avatar

    validates :username, presence: true, uniqueness: true
    validates :email, uniqueness: true, presence: true

    validates_format_of :email, :with => /\A([^@\s]+)@((?:[-a-z0-9]+\.)+
    [a-z]{2,})\z/i

        def User.digest(string)
            cost = ActiveModel::SecurePassword.min_cost ?
            BCrypt::Engine::MIN_COST :
                                                         BCrypt::Engine.
                                                         cost
            BCrypt::Password.create(string, cost: cost)
        end

    end
```



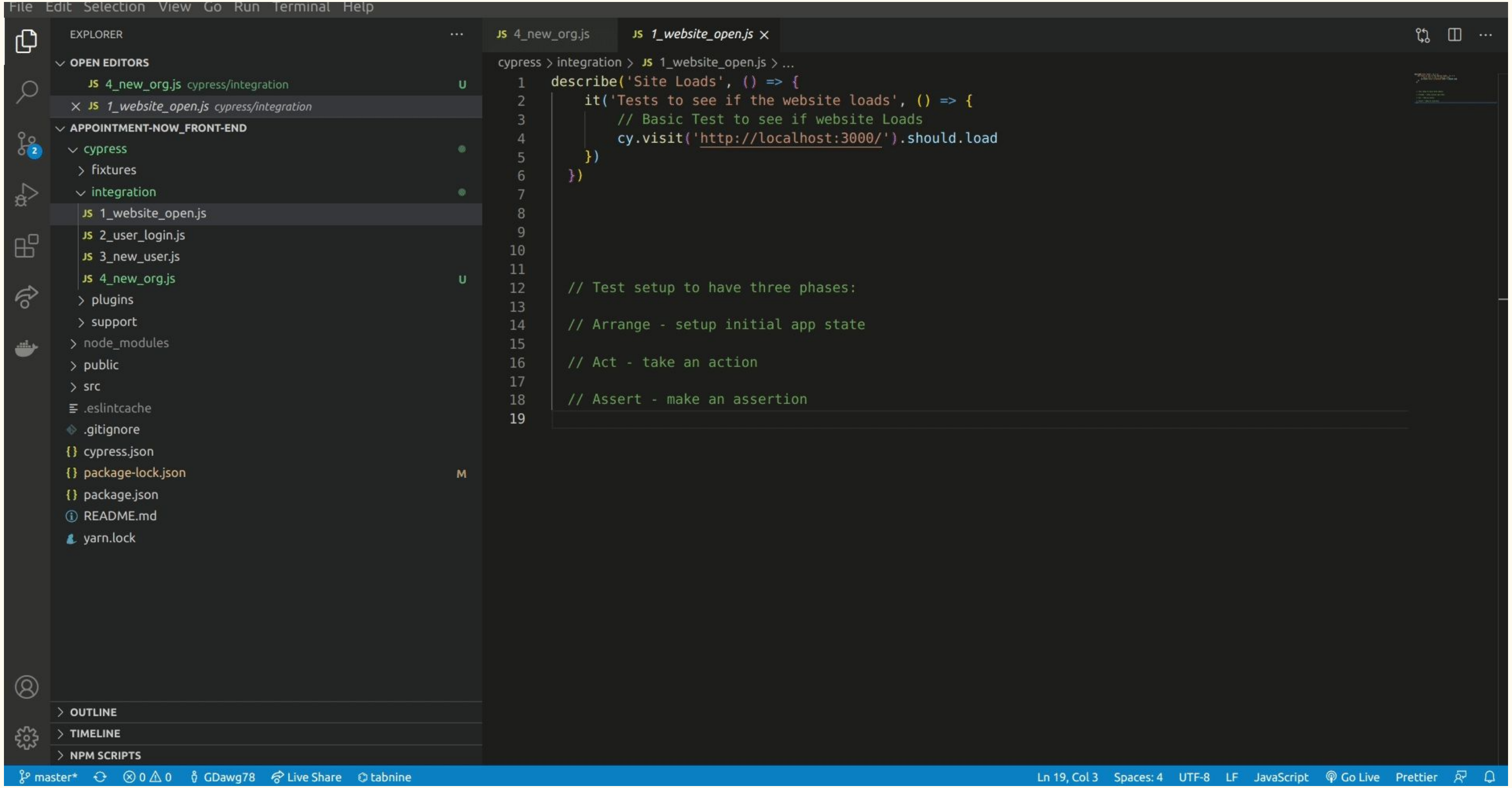
```ruby
def org_routes
  @org_param_routes = []

  Organization.all.each do |org|
    @org_param_routes << [org.organization_name.parameterize, org.id]
  end

  if @org_param_routes
    render json: {
      organizations_routes: @org_param_routes
    }
  end
end
```

Testing



```ruby
    context 'New User' do
        it 'must create a user' do
            1.times do |i|
                @user = User.new(
                    first_name: 'Test',
                    last_name: 'RSpec',
                    mobile: 55555555,
                    email: 'rspec@gmail.com',
                    password_digest: User.digest('password'),
                    username: 'rspec'
                )
                @user.save
                puts "#{i + 1} user created"
            end

                @address = Address.new(
                    address_line_1: '1 Test Way',
                    address_line_2: 'Unit 1',
                    city: 'London',
                    state: 'England',
                    country: 'United Kingdom',
                    postcode: 'W1 9AH',
                    user_id: @user.id
                )
                @address.save
                puts "address for user created"


            expect(User.last.first_name).to eq('Test')
        end
```

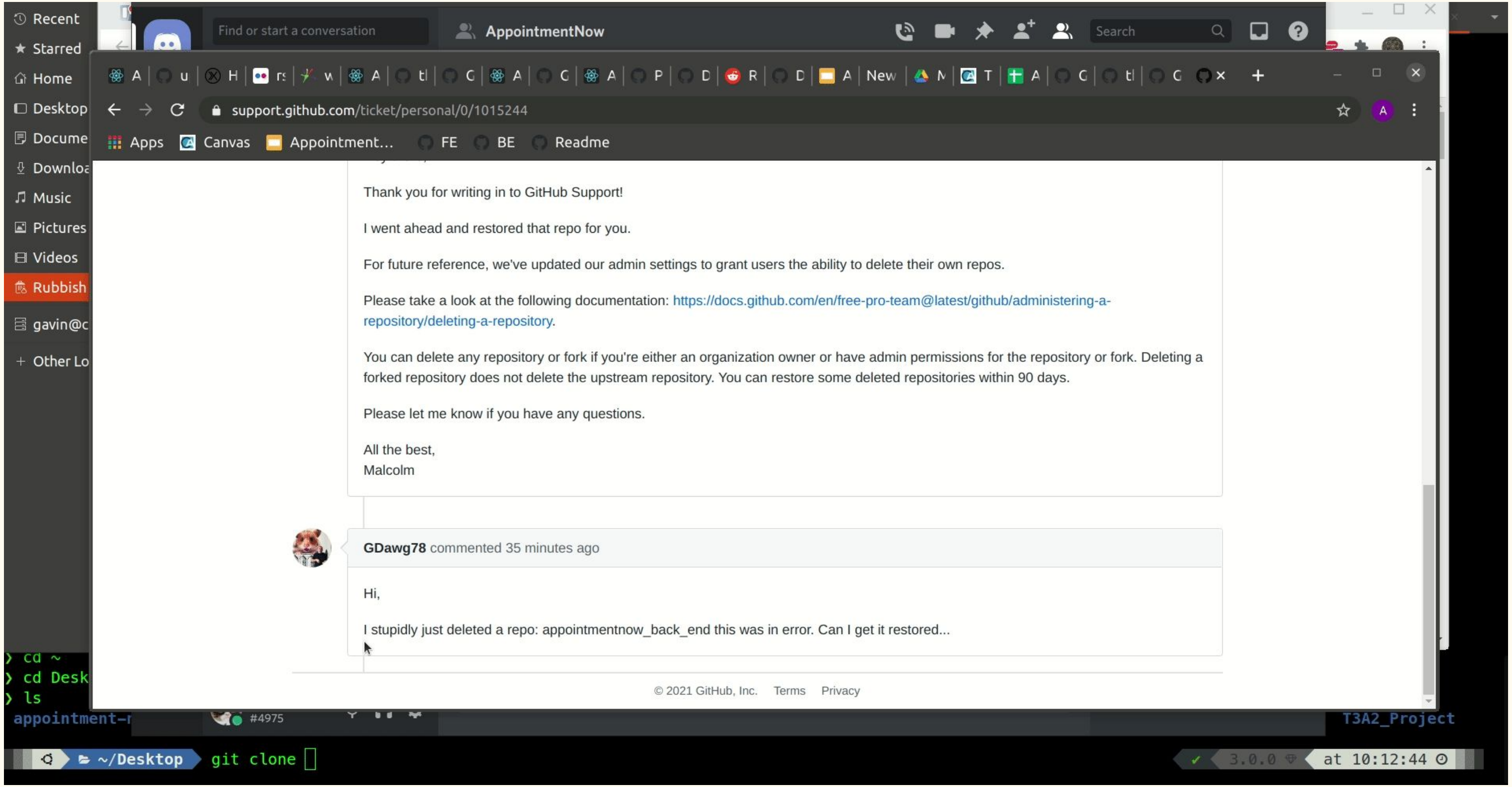
Testing



File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
4_new_org.js cypress/integration U
1_website_open.js cypress/integration
APPOINTMENT-NOW_FRONT-END
cypress
fixtures
integration
1_website_open.js
2_user_login.js
3_new_user.js
4_new_org.js U
plugins
support
node_modules
public
src
.eslintcache
.gitignore
cypress.json
package-lock.json M
package.json
README.md
yarn.lock
OUTLINE
TIMELINE
NPM SCRIPTS
4_new_org.js
1_website_open.js
cypress > integration > 1_website_open.js > ...
master* 0 0 GDawg78 Live Share tabnine
Ln 19, Col 3 Spaces: 4 UTF-8 LF JavaScript Go Live Prettier

```
describe('Site Loads', () => {
    it('Tests to see if the website loads', () => {
        // Basic Test to see if website Loads
        cy.visit('http://localhost:3000/').should.load
    })
  })





  // Test setup to have three phases:

  // Arrange - setup initial app state

  // Act - take an action

  // Assert - make an assertion

```

Testing



Thank you for writing in to GitHub Support!

I went ahead and restored that repo for you.

For future reference, we've updated our admin settings to grant users the ability to delete their own repos.

Please take a look at the following documentation: https://docs.github.com/en/free-pro-team@latest/github/administering-a-repository/deleting-a-repository.

You can delete any repository or fork if you're either an organization owner or have admin permissions for the repository or fork. Deleting a forked repository does not delete the upstream repository. You can restore some deleted repositories within 90 days.

Please let me know if you have any questions.

All the best,
Malcolm

GDawg78 commented 35 minutes ago

Hi,

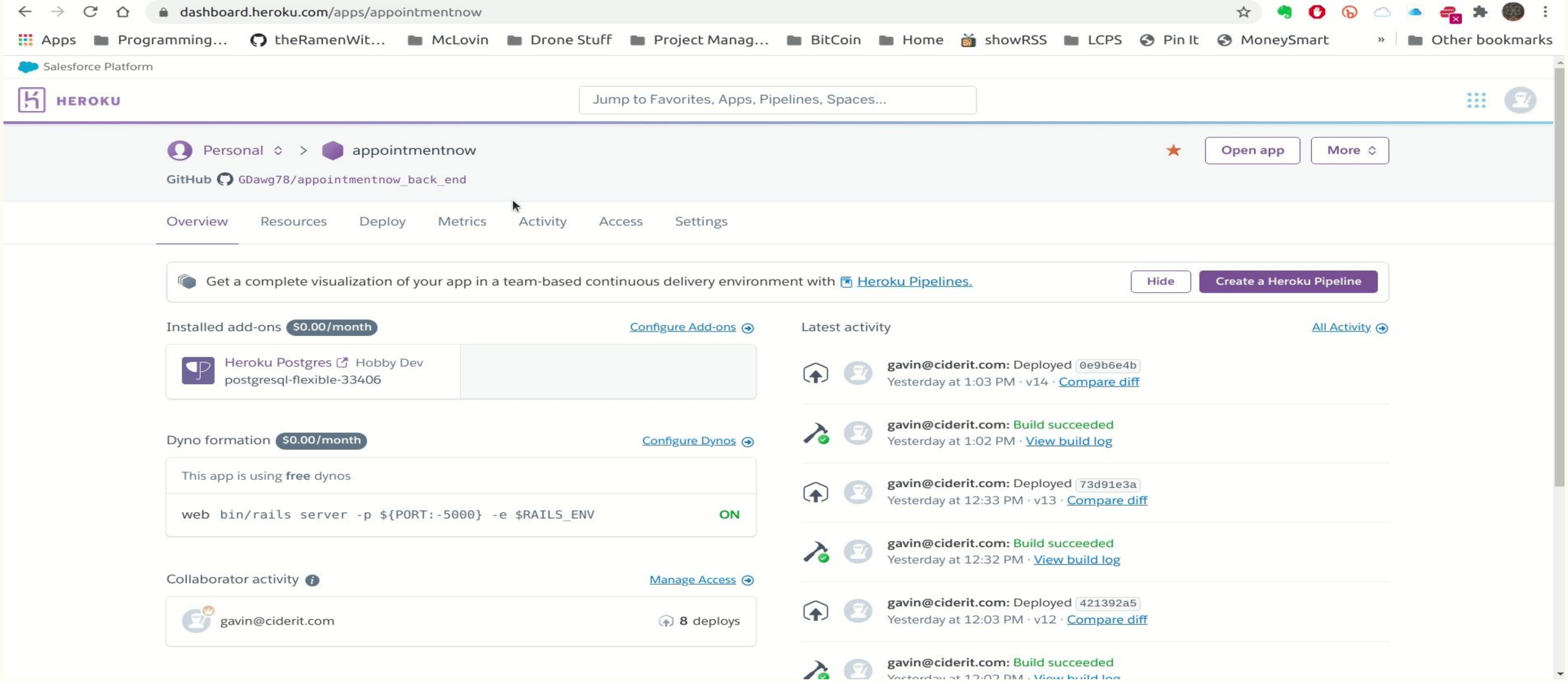
I stupidly just deleted a repo: appointmentnow_back_end this was in error. Can I get it restored...

© 2021 GitHub, Inc. Terms Privacy

```
❯ cd ~
❯ cd Desk
❯ ls
appointment-r
~/Desktop git clone
```

```
✓ 3.0.0 at 10:12:44
```

Hosting



Personal > appointmentnow

GitHub GDawg78/appointmentnow_back_end

Get a complete visualization of your app in a team-based continuous delivery environment with Heroku Pipelines.

Installed add-ons $0.00/month — Configure Add-ons

Heroku Postgres Hobby Dev
postgresql-flexible-33406

Dyno formation $0.00/month — Configure Dynos

This app is using **free** dynos

```
web  bin/rails server -p ${PORT:-5000} -e $RAILS_ENV    ON
```

Collaborator activity — Manage Access

gavin@ciderit.com — 8 deploys

Latest activity — All Activity

gavin@ciderit.com: Deployed `0e9b6e4b`
Yesterday at 1:03 PM · v14 · Compare diff

gavin@ciderit.com: Build succeeded
Yesterday at 1:02 PM · View build log

gavin@ciderit.com: Deployed `73d91e3a`
Yesterday at 12:33 PM · v13 · Compare diff

gavin@ciderit.com: Build succeeded
Yesterday at 12:32 PM · View build log

gavin@ciderit.com: Deployed `421392a5`
Yesterday at 12:03 PM · v12 · Compare diff

gavin@ciderit.com: Build succeeded
Yesterday at 12:02 PM · View build log

Demo