RGB codes:

* Blue: #679AFD
* Pink: #FD7EA7
* Orange: #F7BF48
* Red: #EC4F26
* Green: #17CE71

Main objects of project:

* Students must be able to apply for an apartment
* Manage students and bill payers in apartments
* Manage application? - would require login

Additional:

* a facility to manage rules of the community and issue fines for transgressions
* a way of managing events that occur in the community
* a way of managing and assigning parking bays in the community

ERD:

USER:

* Email
* Password
* Create own username

STUDENT table

* STU\_ID (PRIMARY KEY)
* STU\_NAME
* STU\_EMAIL
* STU\_PHONE
* STU\_STUDY\_PROG
* STU\_STUDY\_YEAR
* STU\_LEASE\_YEAR
* BUILD\_ID (FOREIGN KEY)
* Parking bay needed (boolean)

BILL PAYER table

* STU\_ID (FOREIGN KEY) (PRIMARY KEY)
* BP\_ID (PRIMARY KEY)
* BP\_NAME
* BP\_EMAIL
* BP\_PHONE

RESIDENT table

* APP\_ID (status as residence determined by whether applicant is accepted or not)
* RES\_ID

BUILDING table

* BUILD\_ID (PRIMARY KEY)
* BUILD\_ADDRESS
* BUILD\_NAME

APARTMENT table

* BUILD\_ID (FOREIGN KEY) (PRIMARY KEY)
* APART\_ID (PRIMARY KEY)
* APART\_STATUS
* APART\_NUMROOMS

ROOM table

* APART\_ID (FOREIGN KEY) (PRIMARY KEY)
* ROOM\_ID (PRIMARY KEY)
* ROOM\_STATUS

APPLICATION table

* STU\_ID/BP\_ID (FOREIGN KEY) (PRIMARY KEY)
* APP\_ID (PRIMARY KEY)
* APP\_STATUS

FINE table

PARKING table

Activity diagram

Student submits application -> Approve or reject application -> if reject, send email -> if accept, inform and assign room

NPMJS code Library:

Form Maker

<https://www.npmjs.com/package/mobx-react-form>

Other option for Form Maker

<https://www.npmjs.com/package/react-form-management>

React-admin tutorials:

<https://marmelab.com/blog/2019/01/17/react-timeline.html>

<https://marmelab.com/react-admin/AdvancedTutorials.html>

Routing

<https://marmelab.com/react-admin/Routing.html>

MUI Icon Materials

<https://mui.com/material-ui/material-icons/?query=load>

Old Code Files:

adminForm.jsx

import React from 'react';

const AdminForm = () => {

return (

<form>

<label>

Option 1 Form

<input type="text" placeholder="Enter data for Admin" />

</label>

</form>

);

};

export default AdminForm;

applicationForm.jsx

import React from 'react';

const ApplicationForm = () => {

return (

<form>

<label>

<h1>Application Form</h1>

<label STU\_NAME="Name"> First Name:</label>

<input type="text" placeholder="First Name"/>

<br/>

<label STU\_MNAME="Middle Name"> Middle Name:</label>

<input type="text" placeholder="Middle Name"/>

<br/>

<label STU\_LNAME="Surname"> Surname:</label>

<input type="text" placeholder="Surname"/>

<br/>

<label STU\_EMAIL="Email"> Email:</label>

<input type="email" placeholder="Email"/>

<br/>

<label STU\_PHONE="PhoneNumber"> Phone number:</label>

<input type="tel" placeholder="0123456789"/>

<br/>

<label STU\_STUDY\_PROG="Study Program"> Study Program:</label>

<input type="text" placeholder="Bcom Management Science"/>

<br/>

<label STU\_STUDY\_YEAR="Study Year"> Year of Study:</label>

<input type="year" placeholder="First Year"/>

<br/>

<label STU\_LEASE\_YEAR="Lease year"> Lease Year:</label>

<input type="year" placeholder="2023"/>

<br/>

</label>

</form>

);

};

export default ApplicationForm;

existingStudent.jsx

import React from 'react';

const ExistingStudents = () => {

return (

<form>

<label>

Existing students

<input type="text" placeholder="Enter data for Application" />

</label>

</form>

);

};

export default ExistingStudents;

roles.jsx

import React, { useState } from 'react';

import ApplicationForm from './applicationForm';

import AdminForm from './adminForm';

import ExistingStudents from './existingStudent';

import './roles.css';

function SelectRole() {

// Create states to manage checkbox values

const [selectedBox, setSelectedBox] = useState('');

const [showRole, setShowRole] = useState(true);

const [formVisible, setFormVisible] = useState(false);

const [showApplyForm, setShowApplyForm] = useState(false);

// handle radio selection

const handleRadioChange = (event) => {

setSelectedBox(event.target.value);

};

// compulsory submit button selection

const handleSubmit = (event) => {

event.preventDefault();

setFormVisible(true);

setShowRole(false);

};

const handleApply =(event) => {

event.preventDefault();

setShowApplyForm(true);

setShowRole(false);

};

// handle box selection

// const handleCheck = (event) => {

// event.preventDefault();

// console.log('submitted:', selectedBox);

// };

return (

<div className = "container">

{showRole ? (

<div>

<form onSubmit={handleSubmit} className="welcome">

<h1>Welcome!</h1>

<h3> Who are you? </h3>

<label>

<input

type="radio"

name="role"

value="student"

checked={selectedBox === 'student'}

onChange={handleRadioChange}

className="checkRole"

/>

Student

</label>

{selectedBox=== "student" && (

<div>

<label htmlFor="login"> Login:</label>

<input type="text" placeholder="email"/>

<br/>

<input type="text" placeholder="password"/>

<br/>

<button className="submitButton" value="Submit">Submit</button>

<br/>

<p>New applicant? <button className="applyButton" value="Apply" onClick={handleApply}>Apply here!</button></p>

</div>

)}

<br />

<label>

<input

type="radio"

name="role"

value="admin"

checked={selectedBox === 'admin'}

onChange={handleRadioChange}

className="checkRole"

/>

Admin

</label>

{selectedBox=== "admin" && (

<div>

<label htmlFor="login"> Login:</label>

<input type="text" placeholder="email"/>

<br/>

<input type="text" placeholder="password"/>

<br/>

<button className="submitButton" value="Submit">Submit</button>

</div>

)}

</form>

</div>

) : (

<div>

{/\*opens different jsx files for each option\*/}

{formVisible && (

<div>

{selectedBox === 'student' && <ExistingStudents />}

{selectedBox === 'admin' && <AdminForm />}

</div>

)}

</div>

)}

{showApplyForm && <ApplicationForm />}

</div>

);

};

export default SelectRole;

roles.css

@import url('https://fonts.googleapis.com/css2?family=Montserrat&display=swap');

body {

font-family: 'Montserrat', Arial, sans-serif;

background-image: url("data:image/svg+xml,%3csvg xmlns='http://www.w3.org/2000/svg' version='1.1' xmlns:xlink='http://www.w3.org/1999/xlink' xmlns:svgjs='http://svgjs.dev/svgjs' width='1500' height='1000' preserveAspectRatio='none' viewBox='0 0 1500 1000'%3e%3cg clip-path='url(%26quot%3b%23SvgjsClipPath1109%26quot%3b)' fill='none'%3e%3ccircle r='50.38' cx='155.46' cy='756.91' fill='rgba(103%2c 154%2c 253%2c 0.5)'%3e%3c/circle%3e%3ccircle r='29.16' cx='606.37' cy='396.1' fill='rgba(253%2c 126%2c 167%2c 1)'%3e%3c/circle%3e%3ccircle r='92.04' cx='684.36' cy='779.64' fill='rgba(253%2c 126%2c 167%2c 1)'%3e%3c/circle%3e%3ccircle r='48.345' cx='1330.75' cy='938.9' fill='rgba(103%2c 154%2c 253%2c 0.5)'%3e%3c/circle%3e%3ccircle r='45.48' cx='1032.24' cy='201.25' fill='rgba(103%2c 154%2c 253%2c 1)'%3e%3c/circle%3e%3ccircle r='77.825' cx='185.41' cy='378.45' fill='rgba(253%2c 126%2c 167%2c 0.6)'%3e%3c/circle%3e%3ccircle r='72.395' cx='619.51' cy='344.08' fill='rgba(253%2c 126%2c 167%2c 1)'%3e%3c/circle%3e%3ccircle r='53.96' cx='1186.83' cy='252.54' fill='rgba(253%2c 126%2c 167%2c 1)'%3e%3c/circle%3e%3ccircle r='57.59' cx='661.23' cy='436.63' fill='rgba(103%2c 154%2c 253%2c 1)'%3e%3c/circle%3e%3ccircle r='36.835' cx='234.22' cy='500.67' fill='rgba(103%2c 154%2c 253%2c 0.5)'%3e%3c/circle%3e%3ccircle r='84.02' cx='855.76' cy='681.88' fill='rgba(253%2c 126%2c 167%2c 1)'%3e%3c/circle%3e%3ccircle r='36.11' cx='276.9' cy='825.89' fill='rgba(247%2c 191%2c 72%2c 1)'%3e%3c/circle%3e%3ccircle r='34.495' cx='164.58' cy='494.02' fill='rgba(247%2c 191%2c 72%2c 1)'%3e%3c/circle%3e%3ccircle r='87.295' cx='219.93' cy='667.48' fill='rgba(247%2c 191%2c 72%2c 0.55)'%3e%3c/circle%3e%3ccircle r='88.36' cx='259.02' cy='87.95' fill='rgba(247%2c 191%2c 72%2c 0.55)'%3e%3c/circle%3e%3c/g%3e%3cdefs%3e%3cclipPath id='SvgjsClipPath1109'%3e%3crect width='1500' height='1000' x='0' y='0'%3e%3c/rect%3e%3c/clipPath%3e%3c/defs%3e%3c/svg%3e");

}

/\* Role selection box style \*/

/\* .container {

width: 430px;

height: 520px;

transform: translate(-50%,-50%);

left: 50%;

top: 50%;

} \*/

form {

position: absolute;

transform: translate(-50%,-50%);

}

.welcome{

min-height: 300px;

max-height: 600px;

width: 400px;

overflow-y: auto;

background-color: rgba(255, 254, 254, 0.473);

transform: translate(-50%,-50%);

top: 50%;

left: 50%;

backdrop-filter: blur(10px);

border: 2px solid rgba(255,255,255,0.2);

border-radius: 5px;

box-shadow: 0 0 10px rgba(8,7,16,0.2);

padding: 50px 35px;

}

h1, h3, p {

color: rgb(0, 0, 0);

text-align: center;

}

h3 {

font-size: 1.5rem;

margin-bottom: 20px;

}

label {

display: block;

margin: 5px 0;

font-size: 1.2rem;

color: rgb(0, 0, 0);

}

input[type=radio] {

appearance: none;

background-color: #fff;

width: 15px;

height: 15px;

border: 2px solid #ccc;

border-radius: 2px;

display: inline-grid;

place-content: left;

margin-right: 10px;

padding-top: 1px;

align-items: left;

}

input[type=radio]::before {

content: "";

width: 10px;

height: 10px;

transform: scale(0);

transform-origin: bottom left;

background-color: #fff;

clip-path: polygon(13% 50%, 34% 66%, 81% 2%, 100% 18%, 39% 100%, 0 71%);

}

input[type=radio]:checked::before {

transform: scale(1);

}

input[type=radio]:checked{

background-color: #17CE71;

border: 2px solid #17CE71;

}

.submitButton {

background-color: white;

padding: 10px 20px;

border: none;

border-radius: 5px;

cursor: pointer;

margin: 10px;

}

.submitButton:hover {

background-color: #17CE71;

}

.applyButton {

background: none;

border: none;

color: rgb(0, 0, 0);;

padding: 0;

margin: 0;

text-decoration: underline;

font-size: 16px;

}

.applyButton:hover{

color:rgb(50, 204, 251);

background: none ;

}

import { useState } from 'react';

import { useLogin, useNotify, Notification, Login } from 'react-admin';

import "./login.css"

export const LoginPage = () => {

const [email, setEmail] = useState('');

const [password, setPassword] = useState('');

const login = useLogin();

const notify = useNotify();

const handleSubmit = e => {

e.preventDefault();

login({ email, password }).catch(() =>

notify('Invalid email or password')

);

};

return (

<Login className="customLogin">

<div >

<form onSubmit={handleSubmit}>

<label>Email:</label>

<input

name="email"

type="email"

value={email}

onChange={e => setEmail(e.target.value)}

/>

<label>Password:</label>

<input

name="password"

type="password"

value={password}

onChange={e => setPassword(e.target.value)}

/>

</form>

</div>

</Login>

);

};

// export default LoginPage;

import { BooleanField, Datagrid, DateField, EmailField, List, ReferenceField, TextField, ImageField, TextInput, ImageInput } from 'react-admin';

import { PasswordInput, SelectInput, Create, SimpleForm, Show, Edit, SimpleShowLayout} from 'react-admin';

const equalToPassword = (value, allValues) => {

if (value !== allValues.password) {

return 'The two passwords must match';

}

}

export const UserList = () => (

<List>

<Datagrid rowClick="show">

<ImageField label="avatar" source="avatar.src"/>

<DateField source="created" />

<EmailField source="email" />

<BooleanField source="emailVisibility" />

<TextField source="id" />

<TextField source="name" />

<TextField source="role" />

<DateField source="updated" />

<TextField source="username" />

<BooleanField source="verified" />

</Datagrid>

</List>

);

export const UserShow = () => (

<Show>

<SimpleShowLayout>

<DateField source="avatar" />

<DateField source="created" />

<EmailField source="email" />

<BooleanField source="emailVisibility" />

<TextField source="id" />

<TextField source="name" />

<TextField source="role" />

<DateField source="updated" />

<TextField source="username" />

<BooleanField source="verified" />

</SimpleShowLayout>

</Show>

);

export const UserEdit = () => (

<Edit>

<SimpleForm >

<DateField source="avatar" />

<DateField source="created" />

<EmailField source="email" />

<BooleanField source="emailVisibility" />

<TextField source="id" />

<TextField source="name" />

<TextField source="role" />

<DateField source="updated" />

<TextField source="username" />

<BooleanField source="verified" />

</SimpleForm>

</Edit>

);

export const UserCreate = () => (

<Create disableAuthentication>

<SimpleForm>

<ImageInput source="avatar" accept="image/jpeg">

<ImageField source="src" title="title" />

</ImageInput>

<TextInput source="name" />

<TextInput source="email"/>

<PasswordInput source="password"/>

<PasswordInput source="passwordConfirm" validate={equalToPassword}/>

<SelectInput source="role" choices={[

{ id: 'Admin', name: 'Admin'},

{ id: 'Student', name: 'Student' },

]} />

</SimpleForm>

</Create>

);

import { Datagrid, DateField, List, NumberField, ReferenceField, TextField, Show, SimpleShowLayout, Edit, NumberInput, ReferenceInput, BooleanInput } from 'react-admin';

import { TabbedForm, Create, SimpleForm, TextInput, DateInput, required, SelectInput, CreateButton } from 'react-admin';

import { Box, Button, Typography } from '@mui/material';

import './component.css'

// const Aside = () => (

// <div className='aside' >

// <h6> Post details</h6>

// <p>

// Posts will only be published once an editor approves them

// </p>

// </div>

// );

const Empty = () => (

<Box textAlign="center" m={1} className='empty'>

<Typography variant="h4" paragraph>

No applicantion submitted

</Typography>

<Typography variant="body1">

Create one now!

</Typography>

<CreateButton label="Create applicantion" resource='Application'/>

</Box>

);

export const ApplicationList = () => (

<List empty={<Empty />}>

<Datagrid rowClick="show">

<ReferenceField

label="Application Status"

source="Status"

reference="Application\_Status"

/>

<TextField source="First\_Names" />

<TextField source="Last\_Name" />

<TextField source="Phone\_Number" />

<TextField source="Email" />

<TextField source="Lease\_Year" />

<NumberField source="Student\_Number" />

<TextField source="Study\_Program" />

<TextField source="Study\_Year" />

</Datagrid>

</List>

);

export const ApplicationShow = () => (

<Show>

<SimpleShowLayout>

<TextField source="First\_Names" />

<TextField source="Last\_Name" />

<TextField source="Lease\_Year" />

<TextField source="Email" />

<TextField source="Phone\_Number" />

<NumberField source="Student\_Number" />

<TextField source="Study\_Program" />

<TextField source="Study\_Year" />

<ReferenceField

label="Application Status"

source="Status"

reference="Application\_Status"

/>

</SimpleShowLayout>

</Show>

);

export const ApplicationEdit = () => (

<Edit>

<SimpleForm>

<TextInput source="First\_Names" />

<TextInput source="Last\_Name" />

<TextInput source="Email" />

<TextInput source="Phone\_Number" />

<TextField source="Lease\_Year" />

<NumberInput source="Student\_Number" />

<TextInput source="Study\_Program" />

<SelectInput source="Study\_Year" choices={[

{ id: '1', name: 1 },

{ id: '2', name: 2 },

{ id: '3', name: 3 },

{ id: '4', name: 4 },

]} />

<ReferenceInput

label="Application Status"

source="Status"

reference="Application\_Status"

/>

</SimpleForm>

</Edit>

);

export const ApplicationCreate = () => (

<Create>

<TabbedForm>

<TabbedForm.Tab label="Application"/>

<TextInput source="First\_Names" validate={[required()]} />

<TextInput source="Last\_Name" validate={[required()]} />

<TextInput source="Email" validate={[required()]} />

<TextField source="Phone\_Number" validate={[required()]} />

<TextInput source="Study\_Program" />

<SelectInput source="Study\_Year" choices={[

{ id: '1', name: 1 },

{ id: '2', name: 2 },

{ id: '3', name: 3 },

{ id: '4', name: 4 },

]} />

<SelectInput source="Lease\_Year" choices={[

{ id: '2023/2024', name: '2023/2024' },

{ id: '2024/2025', name: '2024/2025' },

{ id: '2025/2026', name: '2025/2026' },

{ id: '2026/2027', name: '2026/2027' },

]} />

<NumberInput source="Student\_Number" />

<BooleanInput source ="Parking\_Bay" />

<ReferenceInput

label="Application Status"

source="Status"

reference="Application\_Status"

/>

<ReferenceInput

label="Username"

source="User\_id"

reference="Users"

/>

</TabbedForm.Tab>

<TabbedForm.Tab label="Bill Payer">

<TextInput source="First\_Name" validate={[required()]} />

<TextInput source="Last\_Name" validate={[required()]} />

<TextInput source="Email" validate={[required()]} />

<TextInput source="Phone\_Number" validate={[required()]} />

</TabbedForm.Tab>

</TabbedForm>

</Create>

);

export const ApplicationCreate = () => (

<Create>

<TabbedForm>

<TabbedForm.Tab label="Application">

<TextInput source="First\_Names" validate={[required()]} />

<TextInput source="Last\_Name" validate={[required()]} />

<TextInput source="Email" validate={[required()]} />

<TextInput source="ID\_or\_Passport\_Number" />

<TextInput source="Phone\_Number" validate={[required()]} />

<TextInput source="Student\_Number" validate={[required()]} />

<TextInput source="Study\_Program" />

<SelectInput source="Study\_Year" choices={[

{ id: '1', name: 1 },

{ id: '2', name: 2 },

{ id: '3', name: 3 },

{ id: '4', name: 4 },

]} />

<SelectInput source="Lease\_Year" choices={[

{ id: '2023/2024', name: '2023/2024' },

{ id: '2024/2025', name: '2024/2025' },

{ id: '2025/2026', name: '2025/2026' },

{ id: '2026/2027', name: '2026/2027' },

]} />

<TextInput source="Student\_Number" />

<BooleanInput source ="Parking\_Bay" />

<ReferenceInput

label="Application Status"

source="Status"

reference="Application\_Status"

/>

<ReferenceInput

label="Username"

source="User\_id"

reference="Users"

/>

</TabbedForm.Tab>

<TabbedForm.Tab label="Bill Payer">

<TextInput source="First\_Name" validate={[required()]} />

<TextInput source="Last\_Name" validate={[required()]} />

<TextInput source="Email" validate={[required()]} />

<TextInput source="ID\_or\_Passport\_Number" />

<TextInput source="Phone\_Number" validate={[required()]} />

</TabbedForm.Tab>

</TabbedForm>

</Create>

);

**Stopped codespace**

**Comments:**

**Apartments.jsx**

**// Defined a list component for the 'Apartment' resource**

**//Defined a show form for the 'Apartment' resource**

**Application.jsx**

**//Empty component that provides applicants information about the 'apply process'**

**//Defined a list form for 'Application'**

**//Defined a show form for 'Application'**

**//Created a tab form edit form for 'Application'**

**//Created a tab form for 'Application Create’**

**ApplicationStatus.jsx delete**

**Building.jsx**

**// Defined a list component for the 'Building' resource**

**//Defined a show component for 'Building' resource**

**Communication.jsx**

**//Defined a list component for 'Communication'**

**//Defined a show component for 'Communication'**

**Component.css**

**/\* Styling for the login section \*/**

**ContactUs.jsx**

**//Provided contact details for applicants**

**LoginPage.jsx**

Delete comment section

**Parking.jsx**

// created initial screen for applicants who have not been approved yet

//Defined a list form for ‘Parking’

//Defined a show form for ‘Parking’

//Defined a edit form for ‘Parking’ referencing building address

//Defined a create form for ‘Parking’ referencing building address

* Make possible create form for parking

**room .jsx**

//created a generic notice for users who have not been assigned

**Users.jsx**

* Delete commented out code

// created password blah blah blah

**App.jsx**

**/**

**App.jsx**

**// import {} from "./Components/chat";**

**<Resource name="Chat" list={ChatList} show={ChatShow} edit={ChatEdit} create={ChatCreate} />**

**Chat.jsx**

**import { RichTextInput } from 'ra-input-rich-text';**

**For Camryn:**

**Complaint.jsx**

**import { Datagrid, DateField, List, ReferenceField, TextField, Show, SimpleShowLayout, CreateButton, SelectInput, TextInput, Create, SimpleForm, RichTextField } from 'react-admin';**

**import { RichTextInput } from 'ra-input-rich-text';**

**import { Box, Button, Typography } from '@mui/material';**

**import './component.css'**

**const Empty = () => (**

**<Box textAlign="center" m={1} className='emptyPolicy'>**

**<Box className="emptyContent" >**

**<Typography variant="h4" >**

**StelStay Community Policy and Rules**

**</Typography>**

**<Typography paragraph>**

**Welcome to our shared university residence! We are delighted to have you as a resident, and we want to ensure that your stay here is safe, comfortable, and enjoyable. To maintain a harmonious living environment, we have established a set of policies and rules that all residents must adhere to. Please read and understand these guidelines, as they are designed to promote a positive and respectful community.**

**</Typography>**

**<br/>**

**<Typography variant="h6">**

**General Rules**

**</Typography>**

**<Typography align='Left'>**

**1. <b>Respect:</b> Respect for fellow residents, staff, and the property is essential. Treat everyone with courtesy and consideration.**

**</Typography>**

**<Typography align='Left'>**

**2. <b>Quiet Hours:</b> Quiet hours are in effect from 10:00 PM to 8:00 AM. During this time, keep noise to a minimum to ensure a peaceful living environment.**

**</Typography>**

**<Typography align='Left'>**

**3. <b>Guests:</b> Guests are allowed, but they must adhere to the residence's rules. Residents are responsible for the behavior of their guests.**

**</Typography>**

**<Typography align='Left'>**

**4. <b>Safety:</b> Safety is a top priority. Please report any safety concerns or maintenance issues immediately to the residence staff.**

**</Typography>**

**<Typography align='Left'>**

**5. <b>Cleanliness:</b> Keep shared spaces, including kitchens, bathrooms, and common areas, clean and tidy. Regularly dispose of trash in the designated bins.**

**Residence Maintenance**

**</Typography>**

**<Typography align='Left'>**

**6. <b>Furniture and Appliances:</b> Do not move, alter, or remove furniture and appliances from the common areas. If you encounter any maintenance issues, contact residence staff for assistance.**

**</Typography>**

**<Typography align='Left'>**

**7. <b>Decorations:</b> You may decorate your personal space within reason, but do not use nails, screws, or adhesives that may damage walls or surfaces.**

**</Typography>**

**<Typography align='Left'>**

**8. <b>Pets:</b> Pets are not allowed in the residence, with the exception of service animals as permitted by law.**

**</Typography>**

**<br/>**

**<Typography variant="h6">**

**Security**

**</Typography>**

**<Typography align='Left'>**

**9. <b>Keys and Access Cards:</b> Do not share your keys or access cards with others. Report lost or stolen keys/cards immediately.**

**</Typography>**

**<Typography align='Left'>**

**10. <b>Locking Doors:</b> Lock your room and the main entrance when leaving the residence. Do not allow unauthorized persons to enter the building.**

**</Typography>**

**<br/>**

**<Typography variant="h6">**

**Health and Safety**

**</Typography>**

**<Typography align='Left'>**

**11. <b>COVID-19 Guidelines:</b> Adhere to all university and government guidelines related to COVID-19. Maintain good hygiene, wear masks in shared areas when required, and practice social distancing.**

**</Typography>**

**<Typography align='Left'>**

**12. <b>Substances:</b> The use or possession of illegal drugs, including marijuana, is strictly prohibited. The consumption of alcohol is allowed only in accordance with university policies and local laws.**

**</Typography>**

**<br/>**

**<Typography variant="h6">**

**Community Living**

**</Typography>**

**<Typography align='Left'>**

**13. <b>Conflict Resolution:</b> In the event of conflicts with fellow residents, please try to resolve issues through open and respectful communication. If necessary, involve residence staff for mediation.**

**</Typography>**

**<Typography align='Left'>**

**14. <b>Diversity and Inclusion:</b> The residence is committed to diversity and inclusion. Discrimination, harassment, or hate speech will not be tolerated.**

**</Typography>**

**<Typography align='Left'>**

**15. <b>Internet Usage:</b> Respect bandwidth limits and use the internet responsibly. Do not engage in any illegal online activities.**

**</Typography>**

**<br/>**

**<Typography variant="h6">**

**Emergency Procedures**

**</Typography>**

**<Typography align='Left'>**

**16. <b>Emergency Evacuation:</b> Familiarize yourself with the residence's emergency evacuation procedures, and participate in regular fire drills.**

**</Typography>**

**<Typography align='Left'>**

**17. <b>First Aid:</b> In case of injuries or medical emergencies, contact the residence staff and seek assistance immediately.**

**</Typography>**

**<br/>**

**<Typography variant="h6">**

**Penalties for Violation**

**</Typography>**

**<Typography>**

**Residents found in violation of these policies and rules may face disciplinary actions, including warnings, fines, or eviction, depending on the severity of the offense.**

**By choosing to reside in this shared residence, you agree to comply with these policies and rules. Ignorance of these guidelines is not an excuse for non-compliance. Failure to adhere to these rules may result in fines, additional consequences as determined by residence staff and, in extreme cases, eviction.**

**<br/>**

**If you feel that your fellow residents are not adhering to these guidelines, please issue a formal complaint below to be investiagted by community admins.**

**</Typography>**

**<CreateButton label="Submit Complaint" resource='Complaint' sx={{fontSize:'20px', color: '#FD7EA7'}}/>**

**<Typography paragraph>**

**We hope you have a pleasant and productive time in our shared university residence. If you have any questions or concerns, do not hesitate to contact the residence staff. Thank you for being a part of our community!**

**</Typography>**

**</Box>**

**</Box>**

**);**

**export const ComplaintList = () => (**

**<List title="Complaintint" sort={{ field: 'Status', order: 'DESC' }}empty={<Empty />} emptyWhileLoading>**

**<Datagrid rowClick="show">**

**<TextField source="Subject" />**

**<RichTextField source="Complaint"/>**

**<TextField source="Status" />**

**</Datagrid>**

**</List>**

**);**

**export const ComplaintShow = () => (**

**<Show>**

**<SimpleShowLayout>**

**<TextField source="Subject" />**

**<RichTextField source="Complaint"/>**

**<TextField source="Status" />**

**</SimpleShowLayout>**

**</Show>**

**);**

**export const ComplaintEdit = () => (**

**<Edit>**

**<SimpleForm>**

**<TextInput source="Subject" />**

**<RichTextInput source="Complaint"/>**

**<SelectInput source="Status" choices={[**

**{id: 'Resolved' , name: 'Resolved' },**

**{id: 'Unresolved', name: 'Unresolved' },**

**]} />**

**</SimpleForm>**

**</Edit>**

**);**

**const postDefaultValue = () => ({ 'Status': 'Unresolved' });**

**export const ComplaintCreate = () => (**

**<Create>**

**<SimpleForm defaultValues={postDefaultValue}>**

**<TextInput source="Subject" />**

**<RichTextInput source="Complaint"/>**

**<SelectInput source="Status" choices={[**

**{id: 'Resolved' , name: 'Resolved' },**

**{id: 'Unresolved', name: 'Unresolved' },**

**]} />**

**</SimpleForm>**

**</Create>**

**);**

**App.jsx**

**import { Admin, Resource, houseLightTheme, houseDarkTheme } from "react-admin";**

**import { PocketBaseProvider } from "./ra-pocketbase";**

**import { ApplicationList, ApplicationShow, ApplicationEdit, ApplicationCreate } from "./Components/application";**

**import { UserCreate, UserEdit, UserList, UserShow } from "./Components/users";**

**import { LoginPage } from './Components/loginPage'**

**import { CommunicationList, CommunicationShow } from "./Components/communication";**

**import { RoomCreate, RoomEdit, RoomList, RoomShow} from "./Components/room";**

**import { ParkingList, ParkingShow, ParkingEdit } from "./Components/parking";**

**import { ContactList } from "./Components/contactUs";**

**import { ComplaintCreate, ComplaintEdit, ComplaintList, ComplaintShow } from "./Components/complaint";**

**// Menu Icons**

**import ArticleIcon from '@mui/icons-material/Article';**

**import PersonIcon from '@mui/icons-material/Person';**

**import ApartmentIcon from '@mui/icons-material/Apartment';**

**import DirectionsCarIcon from '@mui/icons-material/DirectionsCar';**

**import ChatIcon from '@mui/icons-material/Chat';**

**import ContactMailIcon from '@mui/icons-material/ContactMail';**

**import GavelIcon from '@mui/icons-material/Gavel';**

**export const pbProvider = PocketBaseProvider('https://miniature-space-happiness-r47gjxjgr44cr67-8090.app.github.dev/');**

**export const App = () => (**

**<Admin dataProvider={pbProvider.dataProvider} authProvider={pbProvider.authProvider} loginPage={LoginPage}**

**theme={houseLightTheme} darkTheme={houseDarkTheme}>**

**<Resource name= "Application" list={ApplicationList} show={ApplicationShow} edit={ApplicationEdit} create={ApplicationCreate} icon={ArticleIcon} options={{ label: 'Application' }} recordRepresentation={(record) => `${record.First\_Names} ${record.Last\_Name}`}/>**

**<Resource name ="Application\_Status" recordRepresentation={(record) => `${record.Status}`}/>**

**<Resource name="Building" icon={ApartmentIcon} options={{ label: 'Building' }} recordRepresentation={(record) => `${record.Building\_Name} , ${record.Building\_Address}`} />**

**<Resource name="Apartment" options={{ label: 'Apartment' }} recordRepresentation={(record) => `${record.Apartment\_Number}`} />**

**<Resource name="Room" list={RoomList} show={RoomShow} edit={RoomEdit} create={RoomCreate} icon={ApartmentIcon} options={{ label: 'Room' }} recordRepresentation={(record) => `${record.Room\_Number}`} />**

**<Resource name="Parking" list={ParkingList} show={ParkingShow} edit={ParkingEdit} recordRepresentation={(record) => `${record.Parking\_Number}, ${record.Parking\_Status}`} icon={DirectionsCarIcon} options={{ label: 'Parking' }} />**

**<Resource name="Communication" list={CommunicationList} show={CommunicationShow} icon={ChatIcon} options={{ label: 'Communication' }}/>**

**<Resource name="Users" list={UserList} show={UserShow} edit={UserEdit} create={UserCreate} recordRepresentation={(record) => `${record.name}`} icon={PersonIcon} options={{ label: 'User profile' }} />**

**<Resource name="Complaint" list={ComplaintList} show={ComplaintShow} edit={ComplaintEdit} create={ComplaintCreate} icon={GavelIcon} options={{ label: 'Community Policies' }}/>**

**<Resource name="Contact\_Us" list={ContactList} icon={ContactMailIcon} options={{ label: 'Contact Us' }} />**

**</Admin>**

**);**

**Users.jsx**

**import { BooleanField, Datagrid, DateField, EmailField, List, ReferenceField, TextField, ImageField, TextInput, ImageInput } from 'react-admin';**

**import { PasswordInput, SelectInput, Create, SimpleForm, Show, Edit, SimpleShowLayout} from 'react-admin';**

**import { pbProvider } from '../App';**

**const equalToPassword = (value, allValues) => {**

**if (value !== allValues.password) {**

**return 'The two passwords must match';**

**}**

**}**

**export const UserList = () => (**

**<List title="User" sort={{ field: 'role', order: 'DESC' }} emptyWhileLoading>**

**<Datagrid rowClick="show">**

**<ImageField source='Avatar\_URL' label ="avatar"/>**

**<TextField source="username" />**

**<TextField source="name" />**

**<EmailField source="email" />**

**<DateField source="created" />**

**</Datagrid>**

**</List>**

**);**

**export const UserShow = () => (**

**<Show>**

**<SimpleShowLayout>**

**<ImageField source='Avatar\_URL' label ="avatar"/>**

**<TextField source="username" />**

**<TextField source="name" />**

**<EmailField source="email" />**

**<DateField source="created" />**

**<DateField source="updated" />**

**</SimpleShowLayout>**

**</Show>**

**);**

**export const UserEdit = () => (**

**<Edit>**

**<SimpleForm >**

**<ImageInput source="avatar" />**

**<TextField source="username" />**

**<TextField source="name" />**

**<EmailField source="email" />**

**</SimpleForm>**

**</Edit>**

**);**

**const postDefaultValue = () => ({ 'role': 'Student' });**

**export const UserCreate = () => {**

**<Create disableAuthentication>**

**<SimpleForm defaultValues={postDefaultValue}>**

**<ImageInput source="avatar" >**

**<ImageField source="src" title=""/>**

**</ImageInput>**

**<TextInput source="username" />**

**<TextInput source="name" />**

**<TextInput source="email"/>**

**<PasswordInput source="password"/>**

**<PasswordInput source="passwordConfirm" validate={equalToPassword}/>**

**</SimpleForm>**

**</Create>**

**};**