/\*Листинг компонента SignUp\*/

<template>

<f7-login-screen id="sign-up-screen">

<f7-view>

<f7-page login-screen>

<f7-login-screen-title>Регистрация</f7-login-screen-title>

<f7-list form>

<f7-list-item>

<f7-label>Имя пользователя</f7-label>

<f7-input :value="email"

@input="email = $event.target.value"

placeholder="Имя пользователя"

type="text"></f7-input>

</f7-list-item>

<f7-list-item>

<f7-label>Пароль</f7-label>

<f7-input :value="password"

@input="password = $event.target.value"

type="text"

placeholder="Пароль"></f7-input>

</f7-list-item>

<f7-list-item>

<f7-label>Подтвердите пароль</f7-label>

<f7-input :value="confirmPassword"

@input="confirmPassword = $event.target.value"

type="text"

placeholder="Подтвердите пароль"></f7-input>

</f7-list-item>

<f7-list-button type="submit"

@click="onSignin">Зарегистрироваться</f7-list-button>

<f7-list-button link="/auth/sign-in">Войти</f7-list-button>

<f7-block-footer>

<p>@BSACTeam</p>

</f7-block-footer>

</f7-list>

</f7-page>

</f7-view>

</f7-login-screen>

</template>

<script>

export default {

data() {

return {

email: "",

password: "",

confirmPassword: ""

};

},

computed: {

user() {

return this.$store.getters["user/user"]; }

},

mounted() {

this.$f7.loginScreen.open("#sign-up-screen");

},

created() {

this.$f7.loginScreen.close("#login-screen");

},

watch: {

user(value) {

if (value !== null && value !== undefined) {

this.$f7router.navigate("/");

this.$f7.loginScreen.close("#sign-up-screen");

}

}

},

methods: {

onSignin() {

this.$store.dispatch("user/signUserUp", {

email: this.email,

password: this.password,

configrmPassword: this.confirmPassword

});

}

}

};

</script>

/\*Листинг компонента SignIn\*/

<template>

<f7-login-screen id="login-screen">

<f7-view>

<f7-page login-screen>

<f7-login-screen-title>Вход</f7-login-screen-title>

<f7-list form>

<f7-list-item>

<f7-label>Имя пользователя</f7-label>

<template>

<f7-login-screen id="sign-up-screen">

<f7-view>

<f7-page login-screen>

<f7-login-screen-title>Регистрация</f7-login-screen-title>

<f7-list form>

<f7-list-item>

<f7-label>Имя пользователя</f7-label>

<f7-input :value="email"

@input="email = $event.target.value"

placeholder="Имя пользователя"

type="text"></f7-input>

</f7-list-item>

<f7-list-item>

<f7-label>Пароль</f7-label>

<f7-input :value="password"

@input="password = $event.target.value"

type="text"

placeholder="Пароль"></f7-input>

</f7-list-item>

<f7-list-item>

<f7-label>Подтвердите пароль</f7-label>

<f7-input :value="confirmPassword"

@input="confirmPassword = $event.target.value"

type="text"

placeholder="Подтвердите пароль"></f7-input>

</f7-list-item>

<f7-list-button type="submit"

@click="onSignin">Зарегистрироваться</f7-list-button>

<f7-list-button link="/auth/sign-in">Войти</f7-list-button>

<f7-block-footer>

<p>@BSACTeam</p>

</f7-block-footer>

</f7-list>

</f7-page>

</f7-view>

</f7-login-screen>

</template>

<script>

export default {

data() {

return {

email: "",

password: "",

confirmPassword: ""

};

},

computed: {

user() {

return this.$store.getters["user/user"];

}

},

mounted() {

this.$f7.loginScreen.open("#sign-up-screen");

},

created() {

this.$f7.loginScreen.close("#login-screen");

},

watch: {

user(value) {

if (value !== null && value !== undefined) {

this.$f7router.navigate("/");

this.$f7.loginScreen.close("#sign-up-screen");

}

}

},

methods: {

onSignin() {

this.$store.dispatch("user/signUserUp", {

email: this.email,

password: this.password,

configrmPassword: this.confirmPassword

});

}

}

};

</script>

/\*Листинг компонента Exercises\*/

<template>

<f7-page>

<f7-navbar title="Упражнения"

back-link="Back"></f7-navbar>

<f7-card>

<f7-card-header>Отжимания</f7-card-header>

<f7-card-content>

<iframe style="width:100%;border:0"

src="https://www.youtube.com/embed/\_U8XRJi\_1KY"

webkitallowfullscreen

mozallowfullscreen

allowfullscreen></iframe>

</f7-card-content>

<f7-card-footer>

Отжимания - главное упражнение для верхней части тела. Оно помогает развить силу

и выносливость, нарастить мышцы, укрепить суставы, и, помимо тренировки мышц

верхней части тела, помогает наладить их согласованную работу с мышцами средней

и нижней частей тела.

</f7-card-footer>

</f7-card>

</f7-page>

</template>

/\*Листинг компонента Exercises\*/

<template>

<f7-page>

<f7-navbar>

<f7-nav-left>

<f7-link icon-if-ios="f7:menu"

icon-if-md="material:menu"

panel-open="left"></f7-link>

</f7-nav-left>

<f7-nav-title>Упражнения</f7-nav-title>

<f7-nav-right>

<f7-link icon-if-ios="f7:search"

icon-if-md="material:search"

@click="search.show=!search.show"></f7-link>

</f7-nav-right>

</f7-navbar>

<f7-searchbar v-show="search.show"

disable-link-text="Отменить"

search-container="#search-list"

placeholder="Найти упражнение"

:clear-button="true"

@input="search.query = $event.target.value"

@searchbar:clear="search.query = ''">

<slot name="after-inner">

<f7-link icon-if-ios="f7:more"

icon-if-md="material:more\_vert"

@click="openSheet"></f7-link>

</slot>

</f7-searchbar>

<f7-card v-for="exercise in filteredExercises"

:key="exercise.title">

<f7-card-header>{{exercise.title}}</f7-card-header>

<f7-card-content>

<img style="width:100%;border:0"

:src="exercise.url" />

</f7-card-content>

<f7-card-actions>

<f7-button @click="$f7router.navigate('/exercises/1')">Посмотреть</f7-button>

</f7-card-actions>

</f7-card>

<search-settings></search-settings>

</f7-page>

</template>

<script>

import SearchSettings from "./SearchSettings.vue";

export default {

data() {

return {

exercises: [

],

search: {

show: false,

query: ""

}

};

},

methods: {

openSheet() {

this.$f7.sheet.open("#settings-sheet", true);

}

},

computed: {

filteredExercises() {

return this.exercises.filter(ex => ex.title.includes(this.search.query));

}

},

components: {

SearchSettings

}

};

</script>

<f7-input :value="email"

@input="email = $event.target.value"

placeholder="Имя пользователя"

type="text"></f7-input>

</f7-list-item>

<f7-list-item>

<f7-label>Пароль</f7-label>

<f7-input :value="password"

@input="password = $event.target.value"

type="text"

placeholder="Пароль"></f7-input>

</f7-list-item>

</f7-list>

<f7-list>

<f7-list-button type="submit"

@click="onSignin">Войти</f7-list-button>

<f7-list-button link="/auth/sign-up">Зарегистрироваться</f7-list-button>

<f7-block-footer>

<p>@BSACTeam</p>

</f7-block-footer>

</f7-list>

</f7-page>

</f7-view>

</f7-login-screen>

</template>

<script>

export default {

data() {

return {

email: "",

password: ""

};

},

computed: {

user() {

return this.$store.getters["user/user"];

}

},

mounted() {

this.$f7.loginScreen.open("#login-screen");

},

created() {

this.$f7.loginScreen.close("#sign-up-screen");

},

watch: {

user(value) {

if (value !== null && value !== undefined) {

this.$f7router.navigate("/");

this.$f7.loginScreen.close("#login-screen");

}

}

},

methods: {

onSignin() {

this.$store.dispatch("user/signUserIn", {

email: this.email,

password: this.password

});

}

}

};

</script>

/\*Листинг компонента SearchSettings\*/

<template>

<f7-sheet id="settings-sheet">

<f7-block-title>Настройки поиска</f7-block-title>

<f7-list>

<f7-list-item checkbox title="По упражнениям" name="demo-checkbox" checked></f7-list-item>

<f7-list-item checkbox title="По критериям" name="demo-checkbox"></f7-list-item>

<f7-list-item checkbox title="По профессиям" name="demo-checkbox"></f7-list-item>

</f7-list>

</f7-sheet>

</template>

<template>

<f7-page>

<f7-navbar title="Предпочтительное время"

back-link="Back"></f7-navbar>

<f7-list block

inset>

<f7-list-item v-for="(item, index) in currentTimes"

:key="index"

@swipeout:overswipe="remove(index)">

<f7-label :attr="{for:index}">{{'Время ' + (index+1)}}</f7-label>

<f7-input type="time"

:attr="{name:index}"

:value="currentTimes[index]"

style="display:inline-block; width:68%"></f7-input>

<f7-button round

color="red"

@click="remove(index)"

style="display:inline-block; width:30%">

<f7-icon f7="delete"></f7-icon>

</f7-button>

</f7-list-item>

</f7-list>

<f7-fab color="pink"

@click="add">

<f7-icon f7="add"></f7-icon>

</f7-fab>

</f7-page>

</template>

<script>

export default {

data() {

return {

times: []

};

},

computed: {

currentTimes() {

var result = this.$store.getters["settings/preferredTainingTime"];

if (result != null) {

this.times = result;

}

return this.times;

}

},

methods: {

add() {

this.times.push("");

},

remove(index) {

this.times.splice(index, 1);

},

save() {

this.$store

.dispatch("settings/changePreferredTrainingTime", this.times)

.then(success => {

this.$router.navigate("/settings");

});

}

}

};

</script>

/\*Листинг компонента PrefferedTime\*/

<template>

<f7-page>

<f7-navbar back-link="Back"

title="Профессия"

sliding></f7-navbar>

<f7-searchbar disable-link-text="Cancel"

search-container="#search-list"

placeholder="Найти профессию"

:clear-button="true"

@input="onSearch($event.target.value)"

:enable="true"

@searchbar:clear="onClear"></f7-searchbar>

<f7-list class="searchbar-not-found">

<f7-list-item title="Nothing found"></f7-list-item>

</f7-list>

<f7-list contacts-list>

<f7-list-group v-for="(group, key) in filtered"

:key="key">

<f7-list-item :title="key"

group-title></f7-list-item>

<f7-list-item v-for="name in group"

:title="name"

:key="name"

:value="name"

:checked="currentProfession.name == name"

@change="setCurrentProfession(id)"

radio></f7-list-item>

</f7-list-group>

</f7-list>

</f7-page>

</template>

<script>

export default {

data: function() {

return {

professions: professions,

filter: "",

profession: {

id: 0,

name: ""

}

};

},

created() {

axios.get("/api/professions/").then(response => {

this.professions = professions;

}, null);

},

computed: {

filtered() {

let result = {};

for (var key in this.professions) {

if (this.professions.hasOwnProperty(key)) {

let items = this.professions[key].filter(profession =>

profession.includes(this.filter)

);

if (items.length > 0) {

result[key] = items;

}

}

}

return result;

},

currentProfession() {

let result = this.$store.getters["settings/profession"];

return result != null ? result : this.profession;

}

},

methods: {

onSearch(query) {

this.filter = query;

},

onClear() {

this.filter = "";

},

setCurrentProfession(value) {

this.profession = value;

}

}

};

var professions = {

A: ["Автогонщик", "Адвокат", "Архитектор"],

Б: ["Блогер", "Ботаник", "Бухгалтер"],

В: ["Вахтер", "Водител", "Военнослужащий", "Врач"],

V: ["Vladimir"]

};

</script>

<template>

<f7-page>

<f7-navbar>

<f7-nav-left>

<f7-link icon-if-ios="f7:menu"

icon-if-md="material:menu"

panel-open="left"></f7-link>

</f7-nav-left>

<f7-nav-title>Настройки</f7-nav-title>

</f7-navbar>

<f7-list>

<f7-list-item link="/settings/profession/"

title="Профессия">

<slot name="after-list">

{{currentProfession.name}}

</slot>

</f7-list-item>

<f7-list-item link="/settings/preferred-time/"

title="Предпочтительное время"></f7-list-item>

</f7-list>

</f7-page>

</template>

<script>

import axios from "axios";

export default {

data() {

return {

profession: {

id: 0,

name: ""

}

};

},

created() {

this.$store.dispatch("settings/getProfession");

},

computed: {

currentProfession() {

let result = this.$store.getters["settings/profession"];

return result != null ? result : this.profession;

}

}

};

</script>

/\*Листинг компонента Profession\*/

<template>

<f7-page>

<f7-navbar title="Предпочтительное время"

back-link="Back"></f7-navbar>

<f7-list block

inset>

<f7-list-item v-for="(item, index) in currentTimes"

:key="index"

@swipeout:overswipe="remove(index)">

<f7-label :attr="{for:index}">{{'Время ' + (index+1)}}</f7-label>

<f7-input type="time"

:attr="{name:index}"

:value="currentTimes[index]"

style="display:inline-block; width:68%"></f7-input>

<f7-button round

color="red"

@click="remove(index)"

style="display:inline-block; width:30%">

<f7-icon f7="delete"></f7-icon>

</f7-button>

</f7-list-item>

</f7-list>

<f7-fab color="pink"

@click="add">

<f7-icon f7="add"></f7-icon>

</f7-fab>

</f7-page>

</template>

<script>

export default {

data() {

return {

times: []

};

},

computed: {

currentTimes() {

var result = this.$store.getters["settings/preferredTainingTime"];

if (result != null) {

this.times = result;

}

return this.times;

}

},

methods: {

add() {

this.times.push("");

},

remove(index) {

this.times.splice(index, 1);

},

save() {

this.$store

.dispatch("settings/changePreferredTrainingTime", this.times)

.then(success => {

this.$router.navigate("/settings");

});

}

}

};

</script>

/\*Листинг компонента Settings\*/

<template>

<f7-page>

<f7-navbar back-link="Back"

title="Профессия"

sliding></f7-navbar>

<f7-searchbar disable-link-text="Cancel"

search-container="#search-list"

placeholder="Найти профессию"

:clear-button="true"

@input="onSearch($event.target.value)"

:enable="true"

@searchbar:clear="onClear"></f7-searchbar>

<f7-list class="searchbar-not-found">

<f7-list-item title="Nothing found"></f7-list-item>

</f7-list>

<f7-list contacts-list>

<f7-list-group v-for="(group, key) in filtered"

:key="key">

<f7-list-item :title="key"

group-title></f7-list-item>

<f7-list-item v-for="name in group"

:title="name"

:key="name"

:value="name"

:checked="currentProfession.name == name"

@change="setCurrentProfession(id)"

radio></f7-list-item>

</f7-list-group>

</f7-list>

</f7-page>

</template>

<script>

export default {

data: function() {

return {

professions: professions,

filter: "",

profession: {

id: 0,

name: ""

}

};

},

created() {

axios.get("/api/professions/").then(response => {

this.professions = professions;

}, null);

},

computed: {

filtered() {

let result = {};

for (var key in this.professions) {

if (this.professions.hasOwnProperty(key)) {

let items = this.professions[key].filter(profession =>

profession.includes(this.filter)

);

if (items.length > 0) {

result[key] = items;

}

}

}

return result;

},

currentProfession() {

let result = this.$store.getters["settings/profession"];

return result != null ? result : this.profession;

}

},

methods: {

onSearch(query) {

this.filter = query;

},

onClear() {

this.filter = "";

},

setCurrentProfession(value) {

this.profession = value;

}

}

};

var professions = {

A: ["Автогонщик", "Адвокат", "Архитектор"],

Б: ["Блогер", "Ботаник", "Бухгалтер"],

В: ["Вахтер", "Водител", "Военнослужащий", "Врач"],

V: ["Vladimir"]

};

</script>

<template>

<div v-show="loading"

id="loading"

class="lds-css">

<div style="width:100%;height:100%"

class="lds-eclipse">

<div></div>

</div>

</div>

</template>

<script>

export default {

data() {

return {};

},

computed: {

loading() {

var loading = this.$store.getters["shared/loading"];

return loading;

}

}

};

</script>

<style scoped>

#loading {

position: fixed;

z-index: 999999;

height: 100%;

width: 100%;

background: #fff;

}

@keyframes lds-eclipse {

0% {

-webkit-transform: rotate(0deg);

transform: rotate(0deg);

}

50% {

-webkit-transform: rotate(180deg);

transform: rotate(180deg);

}

100% {

-webkit-transform: rotate(360deg);

transform: rotate(360deg);

}

}

@-webkit-keyframes lds-eclipse {

0% {

-webkit-transform: rotate(0deg);

transform: rotate(0deg);

}

50% {

-webkit-transform: rotate(180deg);

transform: rotate(180deg);

}

100% {

-webkit-transform: rotate(360deg);

transform: rotate(360deg);

}

}

.lds-eclipse {

position: relative;

}

.lds-eclipse div {

position: absolute;

-webkit-animation: lds-eclipse 1s linear infinite;

animation: lds-eclipse 1s linear infinite;

width: 160px;

height: 160px;

top: 35%;

left: 28%;

border-radius: 50%;

box-shadow: 0 4px 0 0 #1875e5;

-webkit-transform-origin: 80px 82px;

transform-origin: 80px 82px;

}

.lds-eclipse {

width: 100% !important;

height: 100% !important;

position: absolute;

-webkit-transform: translate(-100px, -100px) scale(1) translate(100px, 100px);

transform: translate(-100px, -100px) scale(1) translate(100px, 100px);

}

</style>

<template>

<f7-page>

<f7-navbar title="Упражнение 1 из 3"

back-link="Back"></f7-navbar>

<f7-card>

<f7-card-header>Отжимания</f7-card-header>

<f7-card-content>

<iframe style="width:100%;border:0"

src="https://www.youtube.com/embed/\_U8XRJi\_1KY"

webkitallowfullscreen

mozallowfullscreen

allowfullscreen></iframe>

</f7-card-content>

<f7-card-footer>

Отжимания - главное упражнение для верхней части тела. Оно помогает развить силу

и выносливость, нарастить мышцы, укрепить суставы, и, помимо тренировки мышц

верхней части тела, помогает наладить их согласованную работу с мышцами средней

и нижней частей тела.

</f7-card-footer>

</f7-card>

<f7-button>Выполнено!</f7-button>

</f7-page>

</template>

/\*Листинг компонента Loading\*/

<template>

<f7-page>

<f7-navbar>

<f7-nav-left>

<f7-link icon-if-ios="f7:menu"

icon-if-md="material:menu"

panel-open="left"></f7-link>

</f7-nav-left>

<f7-nav-title>Тренировка</f7-nav-title>

<f7-nav-right>

<f7-link>Начать</f7-link>

</f7-nav-right>

</f7-navbar>

<f7-block v-for="(item, index) in items"

:key="index">

<h3>{{item.header}}</h3>

<ul class="list media-list"

style="padding:0">

<li v-for="exercise in item.exercises"

:key="exercise.title">

<a href="/training"

class="item-link item-content">

<div class="item-media"

style="width:30%"><img :src="exercise.avatar"

style="width:100%" /></div>

<div class="item-inner">

<div class="item-title-row">

<div class="item-title">{{exercise.title}}</div>

</div>

<div class="item-subtitle">{{exercise.countOfRepeats}}</div>

<div class="item-subtitle">Не выполнено</div>

<div class="item-text">{{exercise.subtitle}}</div>

</div>

</a>

</li>

</ul>

</f7-block>

</f7-page>

</template>

<script>

export default {

data() {

return {

items: [

{

header: "Низкая сложность",

exercises: [

{

avatar: "https://www.wikireading.ru/img/143620\_73\_i\_042.png",

title: "Наклоны",

countOfRepeats: 10,

subtitle:

"oчeнь пpocтoe yпpaжнeниe, koтopoe знakomo нam eщe c дeтcтвa"

}

]

},

{

exercises: [

{

avatar:

"https://i.ytimg.com/vi/vJmNWi2EMew/hqdefault.jpg?sqp=-oaymwEYCNIBEHZIVfKriqkDCwgBFQAAiEIYAXAB&rs=AOn4CLBEpvl3mFwNzV-JpynmS5WcO-tSbQ",

title: "Отжиманя",

countOfRepeats: 10,

subtitle:

"Постарайтесь соблюдать технику выполнения. Вдох на во время опускания туловища, выдох на подъеме"

},

{

avatar: "http://sportwiki.to/images/3/3f/Sil\_men\_prised.jpg",

title: "Приседания",

countOfRepeats: 10,

subtitle:

"Одно из базовых силовых упражнений (в том числе в пауэрлифтинге и культуризме)"

}

],

header: "Средняя сложность"

}

]

};

}

};

</script>

<template>

<f7-page>

<h1>How was your training?</h1>

</f7-page>

</template>