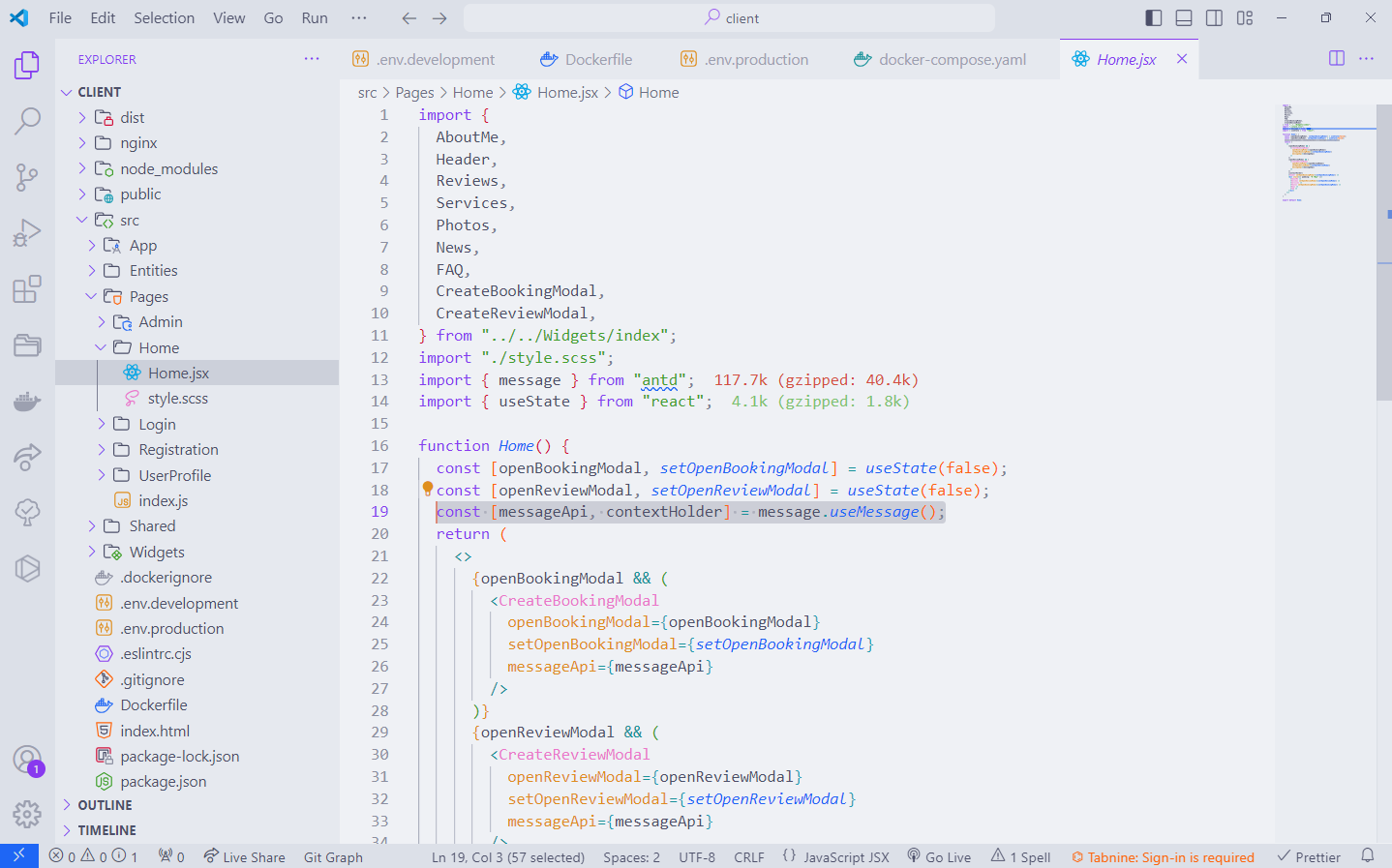
3.2.1

Среда разработки vscode 1.89.0

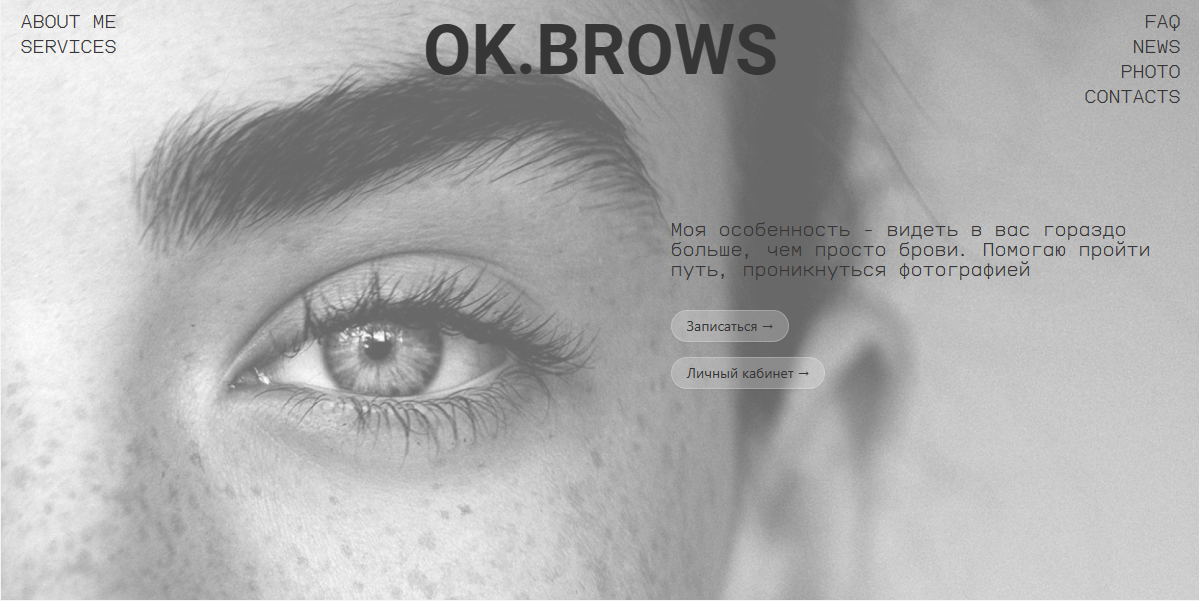


Теги можешь переписать просто

Первая созданная страница «Титульная страница»

Создание проекта происходит выполнением в командной строке команды «npm create vite@latest» после создания проекта в каталоге ./src мы создаем новые страницы выполняя «New File - Создать».

При запуске ПП открывается титульная страница, на которой мы видим ?Якоря(это ссылки которые введут на части страницы хз как это расписать)? Кнопки «Записаться» и «Личный кабинет» если пользователь не авторизован перенаправит на страницу авторизации, иначе «Записаться» откроет модальное окно записи на прием, а «Личный кабинет» откроет страницу с информацией и историей пользователя.



Фрагмент кода для проверки авторизации пользователя, а так же определение роли

const { userData, setUserData } = useContext(UserInfoContext);

const [isAuth, setIsAuth] = useState(!!localStorage.getItem("userData"));

const [isAdmin, setIsAdmin] = useState(false);

useEffect(() => {

if (!!localStorage.getItem("userData")) {

setUserData(JSON.parse(localStorage.getItem("userData")));

if (userData.role === "admin") {

setIsAdmin(true);

setIsAuth(false);

return;

}

setIsAuth(true);

return;

}

}, []);

useEffect(() => {

if (userData.role === "admin") {

setIsAdmin(true);

setIsAuth(false);

return;

}

const isAuth = !Object.values(userData).some((value) => value === "");

setIsAuth(isAuth);

}, [userData]);

const routes = isAdmin ? adminRoutes : isAuth ? authRoutes : unAuthRoutes;

return <RouterProvider router={routes} />;

Далее на титульной странице есть секция «Обо мне»

Изображение выглядит как текст, Человеческое лицо, улыбка, снимок экрана

Автоматически созданное описание

Далее идет секция отзывы, где можно просмотреть отзывы пользователей, а также кнопка «Написать отзыв» которая откроет модальное окно, написания отзыва.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Фрагмент кода создания отзыва

async createReview(client\_name, review\_text, rate) {

await client.query(

`insert into reviews (client\_name, review\_text, rate)

values ($1, $2, $3)`,

[client\_name, review\_text, rate]

);

}

Далее идет секция услуги, где показаны все услуги сайта

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Фрагмент получение услуг и из групп

async getServices() {

const response = await client.query(`SELECT

s.service\_id,

s.service\_name,

s.price,

s.description,

s.service\_group\_id,

sg.service\_group

FROM

services AS s

JOIN

service\_group AS sg ON s.service\_group\_id = sg.service\_group\_id

order by s.service\_id asc`);

return response;

}

Далее идет секция фотографий, где показаны фотографии а также кнопка «Записаться», которая открывает модальное окно для записи.

Изображение выглядит как Человеческое лицо, женщина, снимок экрана, губная помада

Автоматически созданное описание

Фрагмент кода создания записи

async createAppointment(

service\_id,

client\_id,

appointment\_date,

appointment\_time

) {

console.log(service\_id, client\_id, appointment\_date, appointment\_time);

await client.query(

`insert into appointments (

service\_id,

client\_id,

appointment\_date,

appointment\_time

)

values ($1, $2, $3, $4)`,

[service\_id, client\_id, appointment\_date, appointment\_time]

);

}

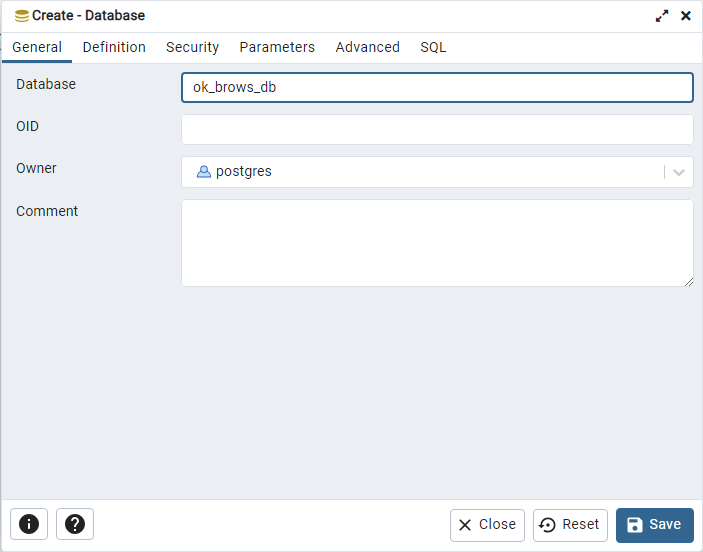
Я описал только то, что видит клиент, я хз админка нужна или нет ну с учетом того, что я без воды описал уже выходит 6 страниц, добавишь теги, описание оформишь все и будет тебе 7-13 страниц как и указано

3.2.2

Используемая бд PosgreSQL, СУБД – PgAdmin

В первом примере PphMyAdmin – это СУБД, MySql – это бд

Создание базы данных «Create - Database» чтобы создать кнопка «Save»



Созданная бд

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Чтобы создать таблицу переходим в NAME\_DATBASE – Schemas – Tables, для создания нажимаем правой кнопкой мыши и выбираем «Create - Table»

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Добавим все поля

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Аналогично и другие

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

3.2.3

Visual studio меняешь Node.js. C# меняешь на JavaScript

Фрагмент кода подключения к бд

import pg from "pg";

import \* as dotenv from "dotenv";

dotenv.config();

const conStringPri = `postgres://${process.env.USERNAME}:${process.env.PASSWORD}@${process.env.HOST}:5432/${process.env.DATABASE}`;

const Client = pg.Client;

const client = new Client({ connectionString: conStringPri });

client.connect();

export default client;

для вывода информации используем тег <table> для ввода информации <input>.

На странице страницу «Услуги» через <input> администратор может добавлять новые услуги.



Фрагмент доавбления услуги

async createService(service\_name, description, price, service\_group\_id) {

await client.query(

`insert into services (service\_name, description, price, service\_group\_id)

values ($1, $2, $3, $4)`,

[service\_name, description, price, service\_group\_id]

);

}

3.2.4

Как в примере

Фрагмент получения заявок

async getAppointments() {

const response =

await client.query(`select a.appointment\_id, a.appointment\_date, a.appointment\_time, s.service\_name, s.price,

ast.status, u.user\_id, u.user\_name, a.service\_id

from appointments as a

inner join services as s on a.service\_id = s.service\_id

inner join appointment\_status as ast on a.status\_id = ast.status\_id

inner join users as u on a.client\_id = u.user\_id

order by a.appointment\_date desc`);

return response;

}

Фрагмент изменения заявок

async editStatusAppointment(appointment\_id) {

await client.query(

`update appointments set

status\_id = 'ce003e07-5888-4df5-85ce-9597b161907e'

where appointment\_id = $1`,

[appointment\_id]

);

const response = await client.query(

`select a.appointment\_date, a.appointment\_time, s.service\_name, s.price, u.user\_name, u.user\_email

from appointments as a

inner join services as s on a.service\_id = s.service\_id

inner join appointment\_status as ast on a.status\_id = ast.status\_id

inner join users as u on a.client\_id = u.user\_id

where a.appointment\_id = $1`,

[appointment\_id]

);

await mailService.sendSuccessfulBooking(response.rows[0]);

}

Фрагмент отправки оповещений на почту

async sendChangeBooking(data) {

await this.transporter.sendMail({

from: process.env.SMTP\_USER,

to: data.user\_email,

subject: "OK.BROWS",

text: "",

html: `

<div>

<h1>Приношу извинения, ваша запись была изменена!</h1>

<p>Новая инфомрация о вашей записи:</p>

<p>Дата записи: ${

data.appointment\_date.toISOString().split("T")[0]

}</p>

<p>Время: ${data.appointment\_time}</p>

<p>Услуга: ${

data.service\_name.charAt(0).toUpperCase() +

data.service\_name.slice(1)

}</p>

<p>Цена: ${data.price} руб.</p>

</div>

`,

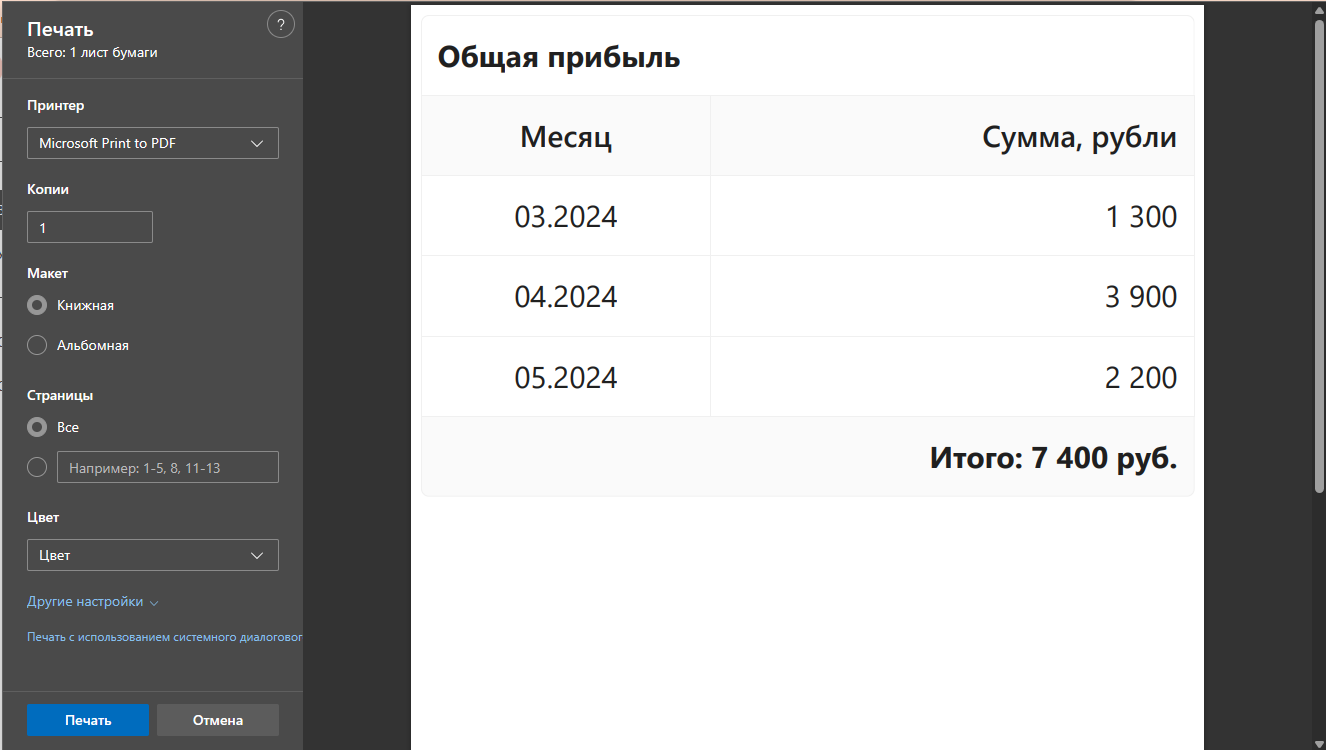
});

}

3.2.5

Для создания отчетов используеться библиотека ReactToPrint которая из html делает pdf

Отчет о прибыли



3.3

Руководство программиста находиться в админке

Изображение выглядит как текст, программное обеспечение, снимок экрана

Автоматически созданное описание

Руководство оператора находиться на главной странице в футере, там кнопка REFERENCE которая открывает pdf в отдельной странице

Изображение выглядит как текст, снимок экрана, программное обеспечение, компьютер

Автоматически созданное описание

3.2.5