Государственное бюджетное профессиональное образовательное учреждение

Ростовской области

|  |
| --- |
| «Таганрогский колледж морского приборостроения» |

|  |
| --- |
| Медицинские услуги. Регистратура. Карточки пациентов, |
| запись на прием к врачу по кабинетами |
| Курсовой проект |
| Пояснительная записка |
| ТКМП МДК.02.01.07.001 ПЗ |

|  |  |
| --- | --- |
| Проверил  *Адардасов А.А.* | Разработал  *Ластовецкий А.М* |

2023

Содержание

[Введение 3](#_Toc138176459)

[1. Постановка задачи 4](#_Toc138176460)

[1.1 Назначение задачи 4](#_Toc138176461)

[1.2 Выбор средств разработки 4](#_Toc138176462)

[1.3 Функциональные характеристики 4](#_Toc138176463)

[1.4 Надежность и безопасность 5](#_Toc138176464)

[1.5 Требования к составу и параметрам технических средств 5](#_Toc138176465)

[2. Проектирование 7](#_Toc138176466)

[3. Программирование 12](#_Toc138176467)

[4. Отладка, оптимизация и тестирование программы 14](#_Toc138176468)

[5. Заключение 15](#_Toc138176469)

[Список используемых источников 16](#_Toc138176470)

[ПРИЛОЖЕНИЕ А. Листинг программы 17](#_Toc138176471)

[ПРИЛОЖЕНИЕ Б. Результаты выполнения программы 35](#_Toc138176472)

# **Введение**

Цель данной курсовой работы заключается в изучении современных подходов к организации работы регистратуры и ведения карточек пациентов в медицинских учреждениях. Данный курсовой проект может быть полезен студентам, изучающим медицинскую тематику, а также работникам медицинских учреждений, которые заинтересованы в обновлении и улучшении своей работы. Исследование этой темы может дать ценные научные выводы и помочь в развитии эффективных стратегий и методов работы регистратуры и ведения карточек пациентов в медицинских учреждениях.

На основе этой цели, были сформулированы следующие задачи:

1. Рассмотреть основные функции регистратуры и процедуры записи на прием по кабинетам.

2. Изучить требования к ведению карточек пациентов в медицинских учреждениях.

3. Рассмотреть организацию работы регистратуры, включая использование электронных систем записи и хранения данных.

4. Проанализировать проблемы и препятствия, связанные с работой регистратуры и ведением карточек пациентов.

5. Изучить возможности для улучшения работы регистратуры и ведения карточек пациентов в медицинских учреждениях.

1. **Постановка задачи**
   1. **Назначение задачи**

В ходе выполнения данного проекта необходимо спроектировать веб-приложение, позволяющее быстро осуществлять запись на прием к врачу.

* 1. **Выбор средств разработки**

Система будет разработана с использованием языка для разработки веб-приложений PHP. Графический интерфейс системы будет реализован при помощи HTML. Все данные будут отображаться на веб-странице в браузере сотрудников.

Готовое решение представляет собой веб-сервис, который можно разместить на любом сервере с поддержкой PHP. Это позволит использовать возможности программы на любых устройствах, имеющих доступ к серверу.

Проект написан в редакторе исходного кода Visual Studio Code.

Для хранения и обращения к данным используется база данных phpMyAdmin.

* 1. **Функциональные характеристики**

При старте в системе должно быть окно обязательной авторизации по уникальному коду и номеру телефона, чтобы остальные пользователи не имели доступ к терминалу.

Веб-сервис должен обеспечивать следующий ряд функций для администратора, доктора и пациента:

* + добавление нового администратора системы;
  + добавление нового доктора в систему;
  + изменение персональных данных доктора;
  + блокировка/снятие блокировки с аккаунта доктора;
  + прием пациента;
  + регистрация;
  + авторизация;
  + редактирование персональных данных пациента;
  + свободный выбор доктора и времени записи на прием;
  + просмотр посещенных приемов у докторов.
  1. **Надежность и безопасность**

Для обеспечения надежной и безопасной работы веб ресурса для администраторов, докторов и пациентов рекомендуются следующие меры:

1. Использование сильных паролей.

2. Установка антивируса и регулярные проверки на наличие вредоносных программ.

3. Ограничение доступа к терминалу только для необходимых пользователей.

4. Шифрование данных, передаваемых через терминал.

5. Регулярные обновления операционной системы и программного обеспечения.

* 1. **Требования к составу и параметрам технических средств**

Программа должна работать на операционных системах семейства Windows, UNIX системах и MacOS; необходимый объем 4 Гб дискового пространства, минимальная мощность процессора должна составлять не менее 1.6 Гц, рекомендуемый объем оперативной памяти должен составлять не менее 4 Гб, расширение монитора 1920x1080.

1. **Проектирование**

В данном разделе описывается процесс разработки веб-приложения, приводится структурная схема и функциональная схема, отражающие состав и взаимодействие по управлению частей разработанного сайта, а также диаграмма вариантов использования и ER-диаграмма.

Решение должно включать в себя модуль обработки данных, модуль интерфейса и модуль для работы с базой данных.

Модуль работы с базой данных включает в себя функции записи и чтения, а также переменные и функции, позволяющие ему обмениваться данными с модулем обработки данных.

Модуль обработки данных получает данные из базы данных, а также из модуля работы с графическим интерфейсом.

Модуль для работы с графическим интерфейсом получает данные, введенные пользователем, и отправляет в модуль обработки данных.

Чтобы лучше понять требования к веб-сайту и определить его структуру, была разработана структурная схема, представленная в соответствии с рисунком 2.1. Данная схема отображает основной набор функций, необходимый для корректной работы администратора и супервайзера.

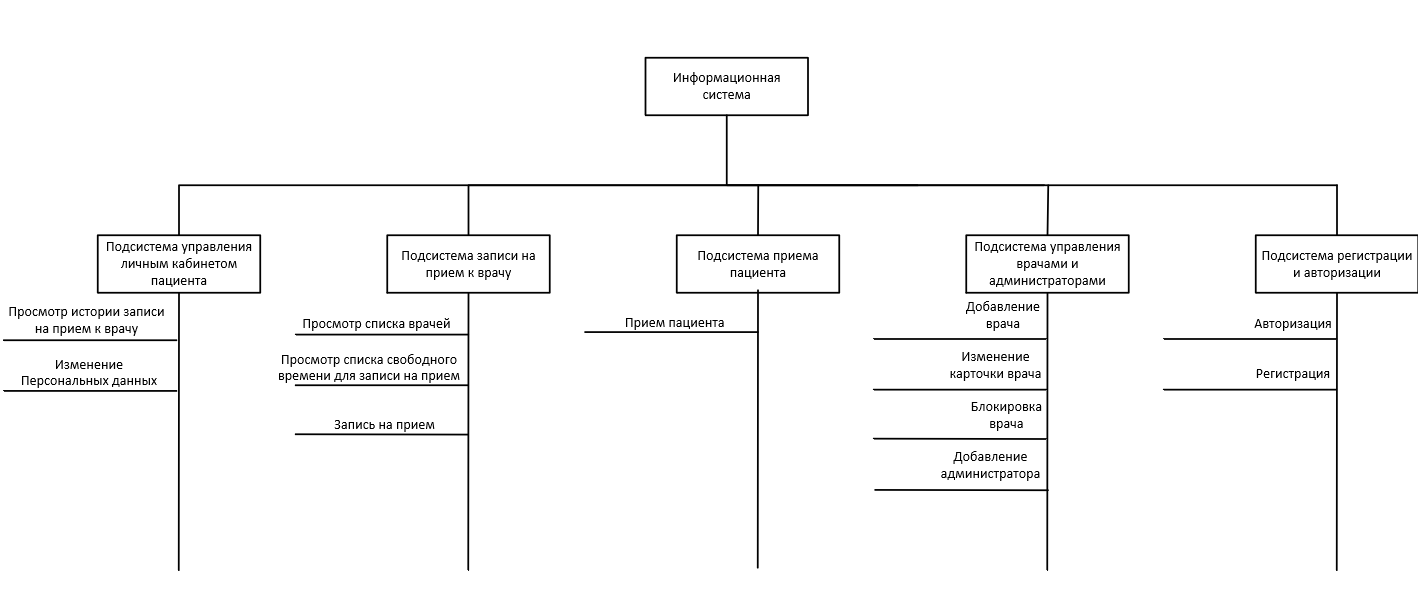
****

Рисунок 2.1 – Структурная схема

Для определения основных функций сайта была разработана функциональная схема, представленная в соответствии с рисунком 2.2. Данная схема отображает процессы, происходящие на разных участках веб-приложения, иллюстрируя в целом весь процесс выполнения.

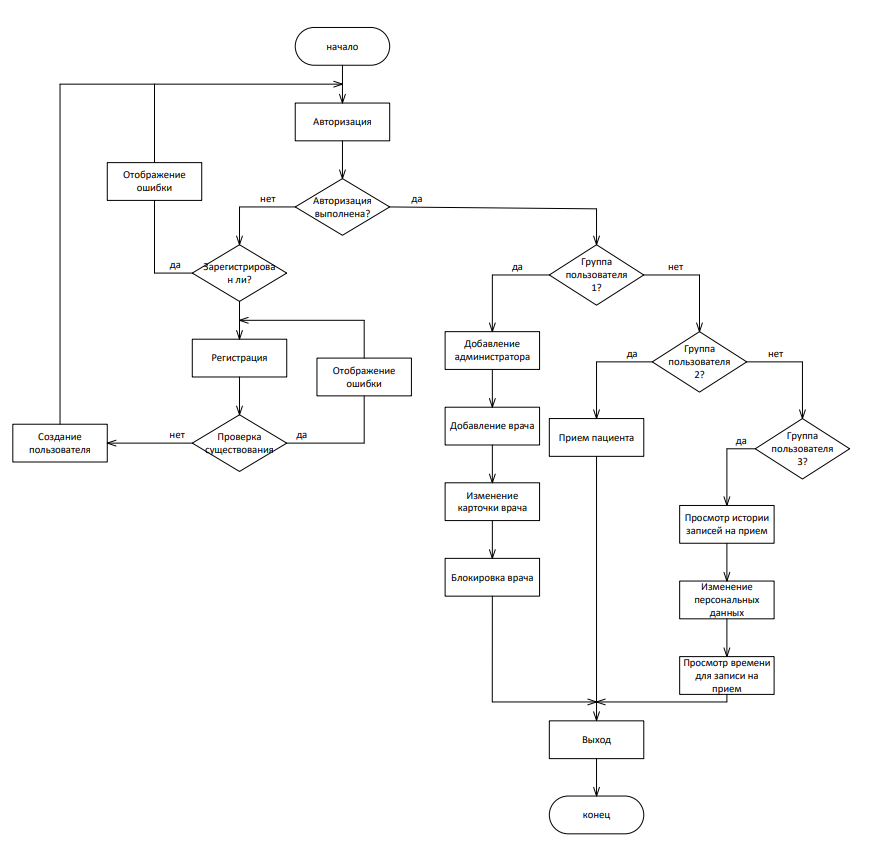


Рисунок 2.2 – Функциональная схема

Проанализировав техническое задание, была построена диаграмма вариантов использования в средстве UML, а также создана модель системы в соответствии с рисунком 2.3

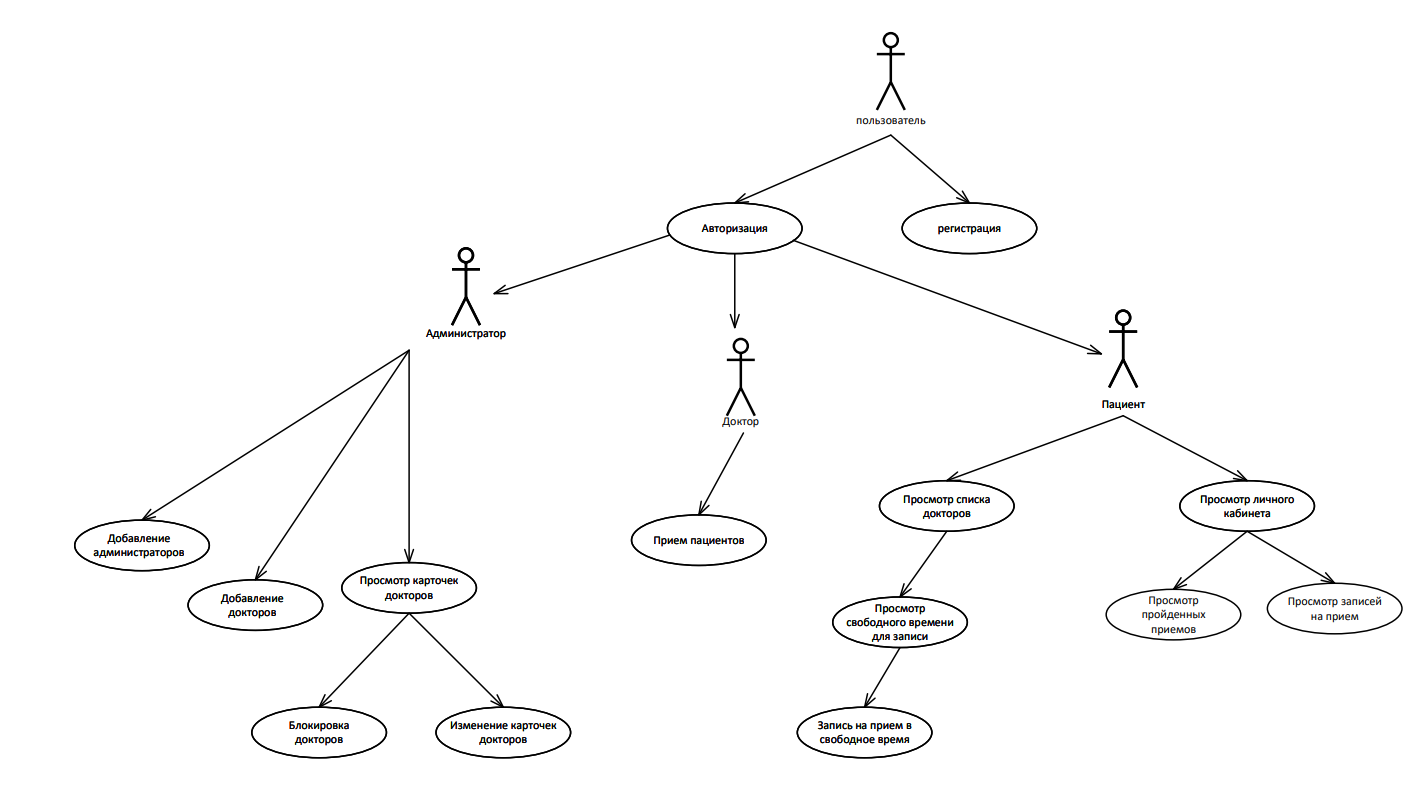


Рисунок 2.3 – Диаграмма вариантов использования

Все актеры должны пройти авторизацию по индивидуальному коду и номеру телефона. Администратор занимается созданием нового аккаунта администратора в системе, созданием нового аккаунта доктора в системе, изменением собственных персональных данных и данных докторов, а также их блокировка и снятие блокировки с аккаунта доктора.

Когда требования определены, можно приступать к проектированию базы данных. На основе исходной информации была разработана ER диаграмма, представленная на рисунке 2.4.

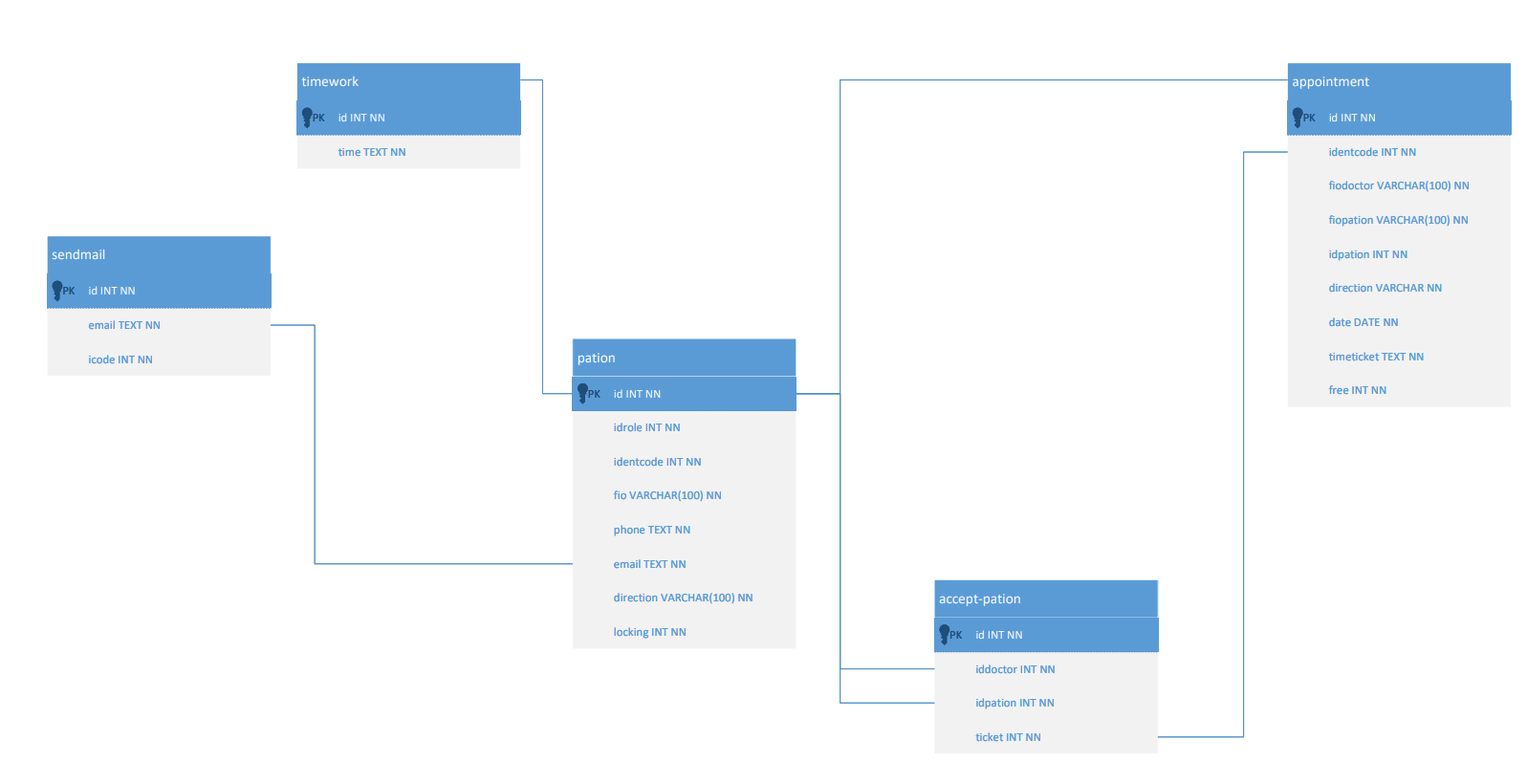


Рисунок 2.4 – ER диаграмма

1. **Программирование**

Функции:

1. mysqli\_connect() – данная функция создает подключение к базе данных MySQL. Она принимает 4 параметра (host – адрес, user – имя пользователя, database – название базы данных, password – пароль от сервера);
2. mysqli\_query() – данная функция выполняет запрос query к базе данных;
3. session\_start() – создаёт сессию, либо возобновляет существующую, основываясь на идентификаторе сессии, переданном через GET- или POST-запрос, либо переданный через cookie;
4. mysqli\_fetch\_assoc – выбирает следующую строку из набора результатов и помещает её в ассоциативный массив;
5. mysqli\_fetch\_all – выбирает все строки из набора результатов и помещает её в ассоциативный массив;

Компонент index.php:

Данный компонент переадресует на страницу авторизации, которая содержит функционал авторизации.

Компонент login.php:

Данный компонент возвращает верстку (структуру элементов) страницы, которая содержит функционал, которая содержит функционал авторизации.

Компонент registration.php:

Данный компонент возвращает верстку (структуру элементов) страницы, которая содержит функционал регистрации.

Компонент admin.php:

Данный компонент возвращает верстку (структуру элементов) страницы, которая содержит функционал администратора системы.

Компонент doctor.php:

Данный компонент возвращает верстку (структуру элементов) страницы, которая содержит функционал доктора в системе.

Компонент pation.php:

Данный компонент возвращает верстку (структуру элементов) страницы, которая содержит функционал пациента в системе.

В данном разделе были описаны функции и компоненты программы.

1. **Отладка, оптимизация и тестирование программы**

Для отладки и тестирования использовался браузер Yandex и встроенные в него инструменты разработчика, а также локальный веб-сервер.

Тестирование:

1. Необходимо запустить локальный веб-сервер;
2. В браузере перейти по адресу «http://localhost/hospital»;
3. Чтобы получить доступ к системе, необходимо ввести правильные индивидуальный код и номер телефона, а иначе пользователя перекинет на страницу “Неверный логин или пароль”;
4. Чтобы добавить нового администратора или доктора в систему, необходимо заполнить форму создания пользователя, которая содержит поля ФИО, номер телефона, E-mail и в случае, если это доктор, то ввести направление доктора.
5. Чтобы изменить свои данные или данные доктора, достаточно в личном кабинете, на главной заменить необходимые поля в форме, а если это доктор, то открыть страницу личного кабинета доктора и там уже заменить персональные данные доктора.
6. Чтобы заблокировать или снять блокировку с аккаунта доктора, достаточно зайти в личный кабинет доктора и нажать кнопку Заблокировать или Разблокировать.
7. Чтобы доктор мог принять пациента, на странице доктора, имеется форма, где доктор ввести индивидуальный код талона, который формируется при записи на прием. Причем доктор может принимать талоны, у которых дата записи соответствует текущей дате.
8. Чтобы записаться на прием к доктору, достаточно просто выбрать доктора, к которому пациент хочет записаться, затем выбрать свободное время, после чего, в личном кабинете пациента, появиться талон с полной информацией о записи к доктору на прием.
9. **Заключение**

В заключение можно подчеркнуть, что тема "Медицинские услуги. Регистратура. Карточки пациентов, запись на приём по кабинетам" является крайне актуальной в нашем современном мире. Обеспечение качественных медицинских услуг становится все более важным в свете развития технологий и увеличения числа населения во всем мире.

Одним из главных преимуществ этой темы является возможность улучшения качества обслуживания пациентов. Автоматизация процесса записи на прием и хранение карточек пациентов позволяет значительно сократить время ожидания и улучшить точность диагностики, что имеет жизненно важное значение в критических случаях.

Также стоит отметить, что данная тема может иметь положительный вклад в борьбу с пандемией COVID-19. Это связано с тем, что уменьшение очередей и ускорение процесса обслуживания на приеме помогает сократить контактные зоны и уменьшить вероятность распространения инфекции.

Таким образом, можно уверенно заключить, что тема "Медицинские услуги. Регистратура. Карточки пациентов, запись на приём по кабинетам" является актуальной и важной не только для улучшения качества медицинских услуг, но и для решения текущих проблем в мире. Автоматизация этого процесса поможет оптимизировать работу медицинских учреждений и обеспечить население качественной медицинской помощью.

# **Список используемых источников**

1 Веллинг Л., Томсон Л. «Разработка веб-приложений с помощью PHP и MySQL».

2 ГОСТ 19.202 – 78. Единая система программной документации. Схемы алгоритмов, программ, данных и систем.

3 Документация по приложению MySQL [Электронный ресурс] – Режим доступа: https://svyatoslav.biz/database\_book\_download/

4 Руководство по Php [Электронный ресурс] – Режим доступа: https://metanit.com/php/tutorial/

5 Учебное пособие по Php [Электронный ресурс] – Режим доступа: https://code.mu/ru/php/book/prime/

6 Фримен Э. «Изучаем HTML, XHTML и CSS».

# **ПРИЛОЖЕНИЕ А. Листинг программы**

Главный файл «index.php»:

<?php

header("Location: ./login/index.php");

?>

Файл подключения к базе данных «db.php»:

<?php

$link = mysqli\_connect("localhost", "hospital", "123456", "hospital");

?>

Файл страницы авторизации «login.php»:

<?php

session\_start();

require\_once("../component/header.php");

?>

<div class="container">

<div class="login-content">

<form action="vendor/accept-phone.php" method="post" id="logform">

<img src="img/avatar.svg">

<h2 class="title">АВТОРИЗАЦИЯ</h2>

<div class="input-div pass">

<div class="i">

<i class="fas fa-lock"></i>

</div>

<div class="div">

<h5>Индивидуальный код</h5>

<input type="text" class="input" name="identcode" id="identcode" target='\_blank'>

</div>

</div>

<div class="input-div pass">

<div class="i">

<i class="fas fa-lock"></i>

</div>

<div class="div">

<h5>Номер телефона</h5>

<input type="text" class="input" name="phone" id="phone" target='\_blank'>

</div>

</div>

<input type="submit" class="btn" value="Войти">

</form>

<div id="errorMes">

<?php

print\_r($\_SESSION["errMes"]);

?>

</div>

<a href="../registration/index.php">Вы не зарагистрированы? Зарегистрироваться</a>

</div>

</div>

<img src="img/foo1.svg" class="img">

<img src="img/foo2.svg" class="img">

<?php

require\_once("../component/footer.php");

?>

Файл страницы регистрации «registration.php»:

<?php

require\_once("../component/header.php");

require\_once("../db/db.php");

?>

<div class="container">

<div class="login-content">

<form action="vendor/registr.php" method="post">

<img src="img/avatar.svg">

<h2 class="title">РЕГИСТРАЦИЯ</h2>

<div class="input-div pass">

<div class="i">

<i class="fa-solid fa-envelope"></i>

</div>

<div class="div">

<h5>E-Mail</h5>

<input type="text" class="input" name="email">

</div>

</div>

<div class="input-div pass">

<div class="i">

<i class="fa-solid fa-phone"></i>

</div>

<div class="div">

<h5>Телефон</h5>

<input type="text" class="input" name="phone">

</div>

</div>

<input type="submit" class="btn" value="Зарегестрироваться">

</form>

</div>

</div>

<img src="img/foo1.svg" class="img">

<img src="img/foo2.svg" class="img">

<?php

require\_once("../component/footer.php")

?>

Файл главной страницы администратора системы «admin.php»:

<?php

session\_start();

if(!isset($\_COOKIE["id"])) {

$\_SESSION["errMes"] = 'Вы не авторизованы, авторизуйтесь!';

header("Location: ../../../../login/index.php");

}

require\_once("../../../../db/db.php");

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap" rel="stylesheet">

<link rel="stylesheet" href="../style/main/style.css">

<link rel="stylesheet" href="../style/main/reset.css">

<link rel="stylesheet" href="../style/main/media.css">

<title>Мой Кабинет | Главная</title>

</head>

<body>

<main>

<header>

<div class="company">

<a href="index.php">Lorem</a>

</div>

<div class="logo"></div>

<button class="header-humburger">

<img src="../img/humburger.png">

</button>

</header>

<aside>

<nav class="nav-menu">

<p class="nm-panel">Панель управления</p>

<ul class="nm-url" style="margin-top: 24px;">

<a href="index.php" class="nm-url-link">Консоль</a>

<a href="create-doctor.php" class="nm-url-link">Добавить Врача</a>

<a href="create-admin.php" class="nm-url-link">Добавить Администратора</a>

</ul>

<ul class="nmu-i" style="margin-top: 24px;">

<a href="index.php" class="nm-url-link nmul-i"><i class="fa-solid fa-house fa-lg"></i></a>

<a href="create-doctor.php" class="nm-url-link nmul-i"><i class="fa-solid fa-user-doctor fa-lg"></i></a>

<a href="create-admin.php" class="nm-url-link nmul-i"><i class="fa-solid fa-hospital-user"></i></a>

</ul>

</nav>

</aside>

<article>

<div class="main-info">

<div class="mi-wrapper">

<div class="my-info">

<div class="adoc-title">

<p>МОИ ДАННЫЕ</p>

</div>

<?php

$idadm = $\_SESSION['iduser'];

$myinfo = mysqli\_query($link, "SELECT \* FROM `pation` WHERE id = '$idadm'");

$myinfo = mysqli\_fetch\_assoc($myinfo);

?>

<form action="vendor/update-profile.php" method="post">

<input type="hidden" name="id" value='<?php echo $myinfo['id']; ?>'>

<div class="text-field text-field\_floating-2">

<input class="text-field\_\_input" type="text" id="fio" name="fio" placeholder="test" value='<?php echo $myinfo['fio']; ?>'>

<label class="text-field\_\_label" for="fio">ФИО</label>

</div>

<div class="text-field text-field\_floating-2">

<input class="text-field\_\_input" type="text" id="phone" name="phone" placeholder="test" value='<?php echo $myinfo['phone']; ?>'>

<label class="text-field\_\_label" for="phone">Телефон</label>

</div>

<div class="text-field text-field\_floating-2">

<input class="text-field\_\_input" type="text" id="email" name="email" placeholder="test" value='<?php echo $myinfo['email']; ?>'>

<label class="text-field\_\_label" for="email">E-Mail</label>

</div>

<input type="submit" class="btn-save" value="Сохранить">

</form>

</div>

<div class="all-doctors">

<div class="adoc-title">

<p>ВСЕ ВРАЧИ</p>

</div>

<div class="adoc-links">

<?php

$doctors = mysqli\_query($link, "SELECT \* FROM `pation` WHERE idrole = 2");

$doctors = mysqli\_fetch\_all($doctors);

foreach($doctors as $doctor) { ?>

<a href="cart-doctor.php?id=<?php echo $doctor[0]; ?>" class="adoc-url"><?php echo $doctor[3]; ?></a>

<?php } ?>

</div>

</div>

<div class="all-direction">

<div class="adoc-title">

<p>ВСЕ НАПРАВЛЕНИЯ</p>

</div>

<div class="adir-list">

<?php

$directions = mysqli\_query($link, "SELECT \* FROM `pation` WHERE idrole = 2 AND id");

$directions = mysqli\_fetch\_all($directions);

foreach($directions as $direction) { ?>

<p><?php echo $direction[6]; ?></p>

<?php } ?>

</div>

</div>

</div>

</div>

</article>

</main>

<div class="mobile-menu">

<div class="header-humburger header-humburger-close">

<img src="img/humburger.png" id="closeMobileNav">

</div>

<nav class="mobile-nav">

<ul>

<li><a href="index.php">Консоль</a></li>

<li><a href="create-doctor.php">Добавить Врача</a></li>

<li><a href="create-admin.php">Добавить Администратора</a></li>

</ul>

</nav>

</div>

<script src="../script/main/humburger.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script src="https://kit.fontawesome.com/61b86703fe.js" crossorigin="anonymous"></script>

</body>

</html>

Файл главной страницы доктора «doctor.php»:

<?php

require\_once("../../../../db/db.php");

session\_start();

if(!isset($\_COOKIE["id"])) {

$\_SESSION["errMes"] = 'Вы не авторизованы, авторизуйтесь!';

header("Location: ../../../../login/index.php");

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap" rel="stylesheet">

<link rel="stylesheet" href="../style/main/style.css">

<link rel="stylesheet" href="../style/main/reset.css">

<link rel="stylesheet" href="../style/main/media.css">

<title>Мой Кабинет | Принять Пациента</title>

</head>

<body>

<main>

<header>

<div class="company">

<a href="index.php">Lorem</a>

</div>

<div class="logo"></div>

<button class="header-humburger">

<img src="../img/humburger.png">

</button>

</header>

<aside>

<nav class="nav-menu">

<p class="nm-panel">Панель управления</p>

<ul class="nm-url" style="margin-top: 24px;">

<a href="index.php" class="nm-url-link">Консоль</a>

</ul>

<ul class="nmu-i" style="margin-top: 24px;">

<a href="index.php" class="nm-url-link nmul-i"><i class="fa-solid fa-house fa-lg"></i></a>

</ul>

</nav>

</aside>

<article>

<div class="main-info">

<div class="mi-wrapper">

<div class="my-info">

<div class="adoc-title">

<p>Принять Пациента</p>

</div>

<form action="vendor/accept-pation.php" method="post">

<div class="text-field text-field\_floating-2">

<input class="text-field\_\_input" type="text" id="idcodeticket" name="idcodeticket" placeholder="test">

<label class="text-field\_\_label" for="fio">Индивидуальный номер талона</label>

</div>

<input type="submit" class="btn-save" value="Принять">

</form>

</div>

</div>

</div>

</article>

</main>

<div class="mobile-menu">

<div class="header-humburger header-humburger-close">

<img src="../img/humburger.png" id="closeMobileNav">

</div>

<nav class="mobile-nav">

<ul>

<li><a href="index.php">Консоль</a></li>

</ul>

</nav>

</div>

<script src="../script/main/humburger.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script src="https://kit.fontawesome.com/61b86703fe.js" crossorigin="anonymous"></script>

</body>

</html>

Файл главной страницы пациента «pation.php»:

<?php

require\_once("../../../../db/db.php");

session\_start();

if(!isset($\_COOKIE["id"])) {

$\_SESSION["errMes"] = 'Вы не авторизованы, авторизуйтесь!';

header("Location: ../../../../login/index.php");

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap" rel="stylesheet">

<link rel="stylesheet" href="../style/main/style.css">

<link rel="stylesheet" href="../style/main/reset.css">

<link rel="stylesheet" href="../style/main/media.css">

<title>Мой Кабинет | Главная</title>

</head>

<body>

<main>

<header>

<div class="company">

<a href="index.php">Lorem</a>

</div>

<div class="logo"></div>

<button class="header-humburger">

<img src="../img/humburger.png">

</button>

</header>

<aside>

<nav class="nav-menu">

<p class="nm-panel">Панель управления</p>

<ul class="nm-url" style="margin-top: 24px;">

<a href="index.php" class="nm-url-link">Консоль</a>

<a href="create-appointment.php" class="nm-url-link">Записаться на прием</a>

</ul>

<ul class="nmu-i" style="margin-top: 24px;">

<a href="index.php" class="nm-url-link nmul-i"><i class="fa-solid fa-house fa-lg"></i></a>

<a href="create-appointment.php" class="nm-url-link nmul-i"><i class="fa-solid fa-user-doctor fa-lg"></i></a>

</ul>

</nav>

</aside>

<article>

<div class="main-info">

<div class="mi-wrapper">

<div class="my-info-2">

<div class="adoc-title">

<p>МОИ ДАННЫЕ</p>

</div>

<?php

$idadm = $\_SESSION['iduser'];

$myinfo = mysqli\_query($link, "SELECT \* FROM `pation` WHERE id = '$idadm'");

$myinfo = mysqli\_fetch\_assoc($myinfo);

?>

<form action="vendor/update-profile.php" method="post">

<input type="hidden" name="id" value='<?php echo $myinfo['id']; ?>'>

<div class="text-field text-field\_floating-2">

<input class="text-field\_\_input" type="text" id="fio" name="fio" placeholder="test" value='<?php echo $myinfo['fio']; ?>'>

<label class="text-field\_\_label" for="fio">ФИО</label>

</div>

<div class="text-field text-field\_floating-2">

<input class="text-field\_\_input" type="text" id="phone" name="phone" placeholder="test" value='<?php echo $myinfo['phone']; ?>'>

<label class="text-field\_\_label" for="phone">Телефон</label>

</div>

<div class="text-field text-field\_floating-2">

<input class="text-field\_\_input" type="text" id="email" name="email" placeholder="test" value='<?php echo $myinfo['email']; ?>'>

<label class="text-field\_\_label" for="email">E-Mail</label>

</div>

<input type="submit" class="btn-save" value="Сохранить">

</form>

</div>

<div class="visiting">

<div class="adoc-title">

<p>ОЖИДАЮТ ПОСЕЩЕНИЯ</p>

</div>

<?php

$visitappoint = mysqli\_query($link, "SELECT \* FROM `appointment` WHERE idpation = '$idadm' AND free = '1' UNION SELECT \* FROM `appointment-2` WHERE idpation = '$idadm' AND free = '1' UNION SELECT \* FROM `appointment-3` WHERE idpation = '$idadm' AND free = '1' UNION SELECT \* FROM `appointment-4` WHERE idpation = '$idadm' AND free = '1' UNION SELECT \* FROM `appointment-5` WHERE idpation = '$idadm' AND free = '1' UNION SELECT \* FROM `appointment-6` WHERE idpation = '$idadm' AND free = '1' UNION SELECT \* FROM `appointment-7` WHERE idpation = '$idadm' AND free = '1' UNION SELECT \* FROM `appointment-8` WHERE idpation = '$idadm' AND free = '1' UNION SELECT \* FROM `appointment-9` WHERE idpation = '$idadm' AND free = '1' UNION SELECT \* FROM `appointment-10` WHERE idpation = '$idadm' AND free = '1'");

$visitappoint = mysqli\_fetch\_all($visitappoint);

?>

<table class="visit-wrapper">

<tr>

<th>Инд. №</th>

<th>ФИО Доктора</th>

<th>ФИО Пациента</th>

<th>Направление доктора</th>

<th>Время Приема</th>

</tr>

<?php foreach($visitappoint as $vt) { ?>

<tr>

<td><?php echo $vt[1]; ?></td>

<td><?php echo $vt[2]; ?></td>

<td><?php echo $vt[3]; ?></td>

<td><?php echo $vt[5]; ?></td>

<td><?php echo $vt[7]; ?></td>

</tr>

<?php } ?>

</table>

</div>

<div class="visiting">

<div class="adoc-title">

<p>УЖЕ ПОСЕЩЕННЫЕ</p>

</div>

<?php

$visitedappoint = mysqli\_query($link, "SELECT \* FROM `appointment` WHERE idpation = '$idadm' AND free = '0' UNION SELECT \* FROM `appointment-2` WHERE idpation = '$idadm' AND free = '0' UNION SELECT \* FROM `appointment-3` WHERE idpation = '$idadm' AND free = '0' UNION SELECT \* FROM `appointment-4` WHERE idpation = '$idadm' AND free = '0' UNION SELECT \* FROM `appointment-5` WHERE idpation = '$idadm' AND free = '0' UNION SELECT \* FROM `appointment-6` WHERE idpation = '$idadm' AND free = '0' UNION SELECT \* FROM `appointment-7` WHERE idpation = '$idadm' AND free = '0' UNION SELECT \* FROM `appointment-8` WHERE idpation = '$idadm' AND free = '0' UNION SELECT \* FROM `appointment-9` WHERE idpation = '$idadm' AND free = '0' UNION SELECT \* FROM `appointment-10` WHERE idpation = '$idadm' AND free = '0'");

$visitedappoint = mysqli\_fetch\_all($visitedappoint);

?>

<table class="visit-wrapper">

<tr>

<th>Инд. №</th>

<th>ФИО Доктора</th>

<th>ФИО Пациента</th>

<th>Направление доктора</th>

<th>Время Приема</th>

</tr>

<?php foreach($visitedappoint as $vted) { ?>

<tr>

<td><?php echo $vted[1]; ?></td>

<td><?php echo $vted[2]; ?></td>

<td><?php echo $vted[3]; ?></td>

<td><?php echo $vted[5]; ?></td>

<td><?php echo $vted[7]; ?></td>

</tr>

<?php } ?>

</table>

</div>

</div>

</div>

</article>

</main>

<div class="mobile-menu">

<div class="header-humburger header-humburger-close">

<img src="../img/humburger.png" id="closeMobileNav">

</div>

<nav class="mobile-nav">

<ul>

<li><a href="index.php">Консоль</a></li>

<li><a href="create-appointment.php">Записаться на прием</a></li>

</ul>

</nav>

</div>

<script src="../script/main/humburger.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script src="https://kit.fontawesome.com/61b86703fe.js" crossorigin="anonymous"></script>

</body>

</html>

# **ПРИЛОЖЕНИЕ Б. Результаты выполнения программы**

Необходимо запустить локальный веб-сервер и перейти по адресу «http://localhost/hospital» при помощи браузера, в окне браузера будет отображён интерфейс в соответствии с рисунком Б.1.

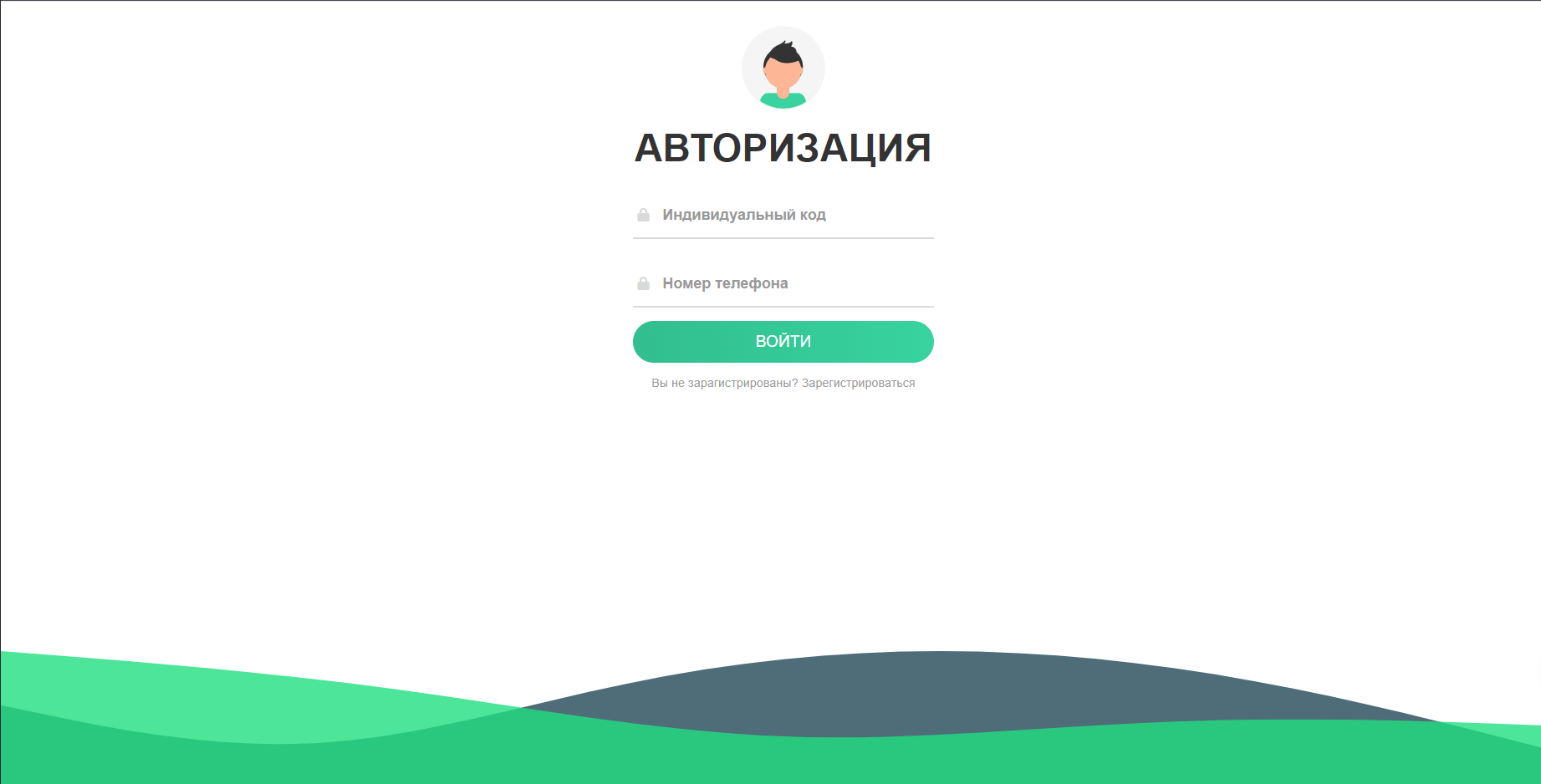


Рисунок Б.1 – Начальное окно сайта

При авторизации, в зависимости от логина и пароля, нам доступен интерфейс для администратора, доктора или пациента в соответствии с рисунками Б.2 – Б.3 – Б4.

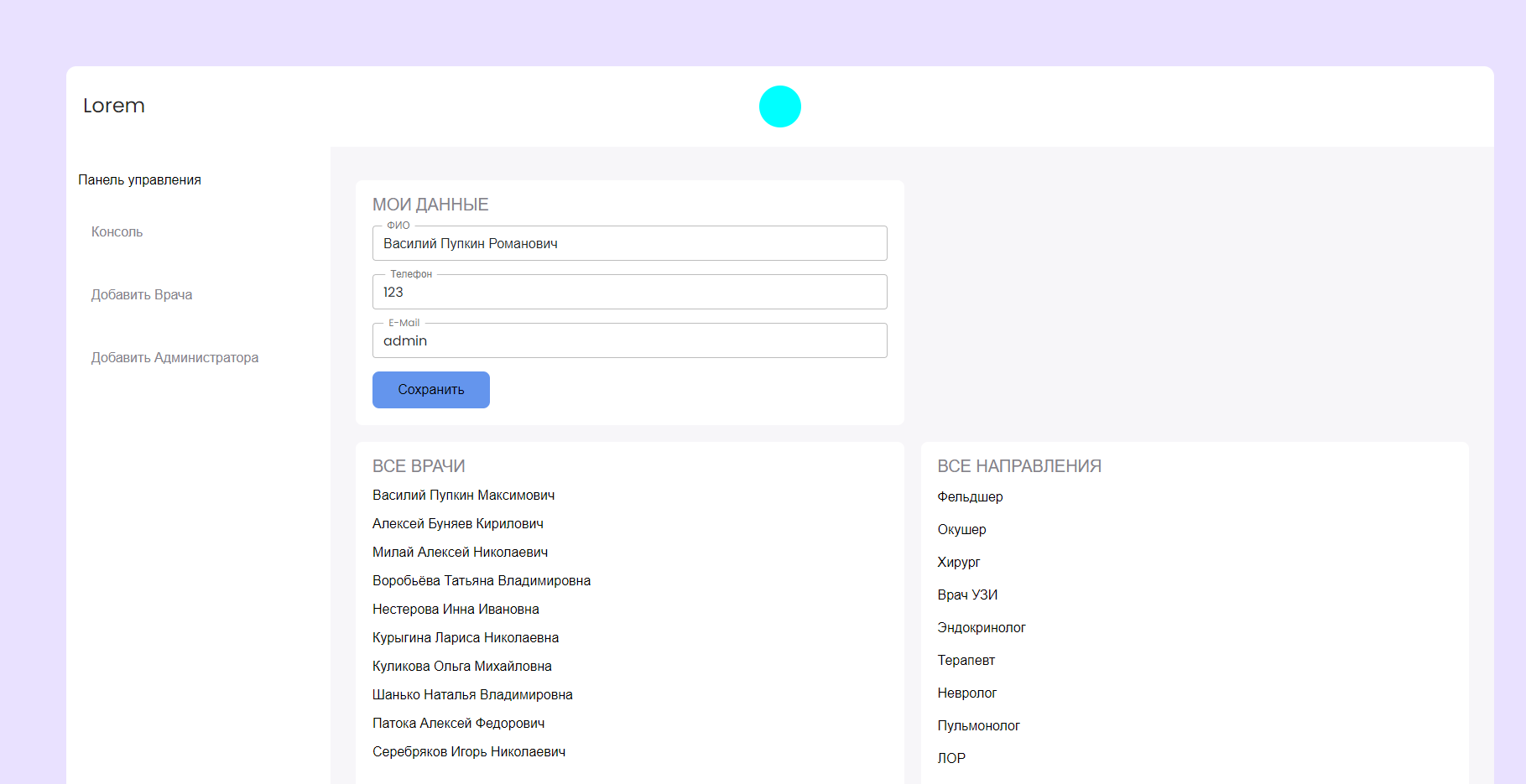


Рисунок Б.2 – Панель управления администратора

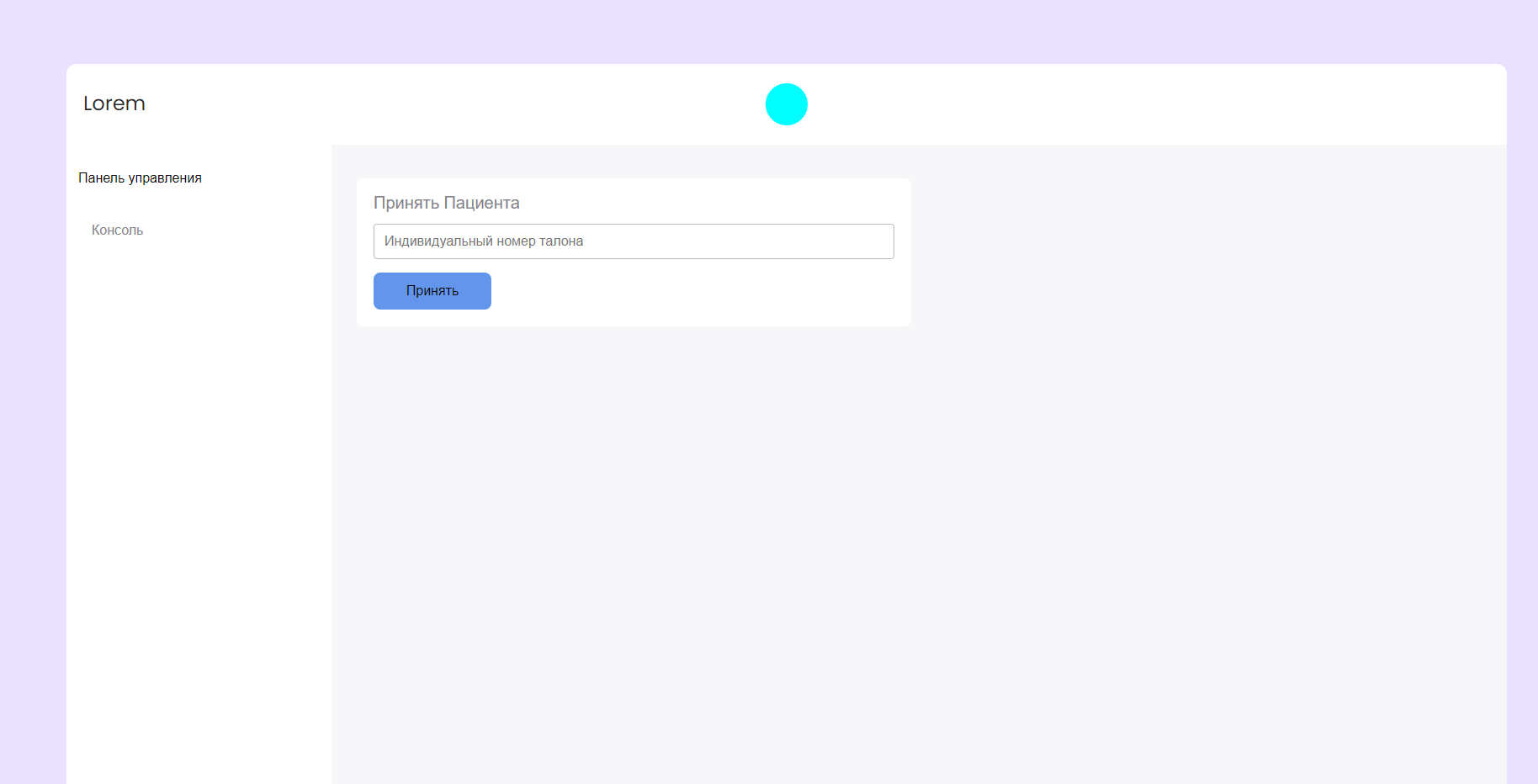


Рисунок Б.3 – Панель управления доктора

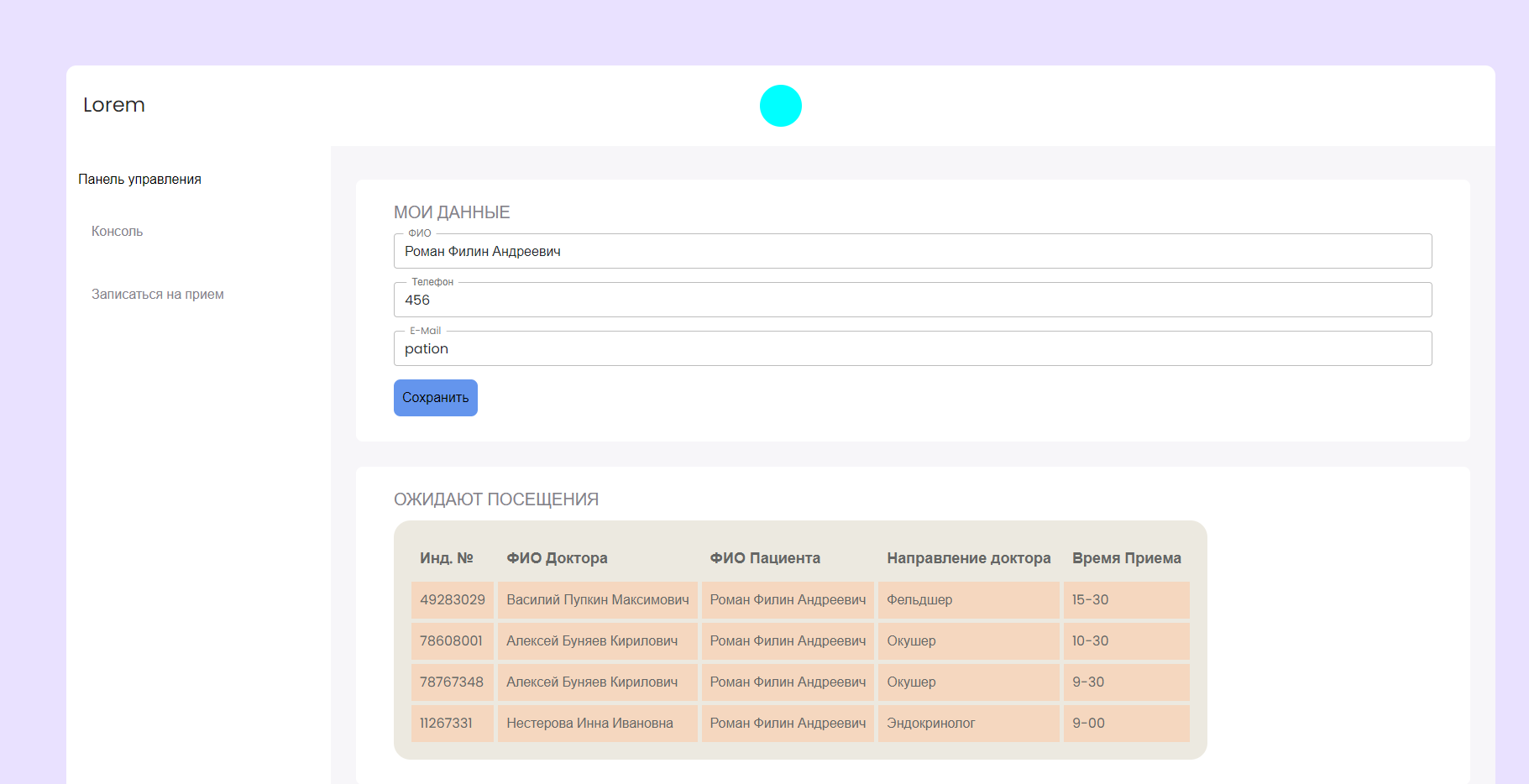


Рисунок Б.4 – Панель управления пациента

При неправильно введенных данных при авторизации пользователя ждет оповещение на другой странице как представлено на рисунке Б.5

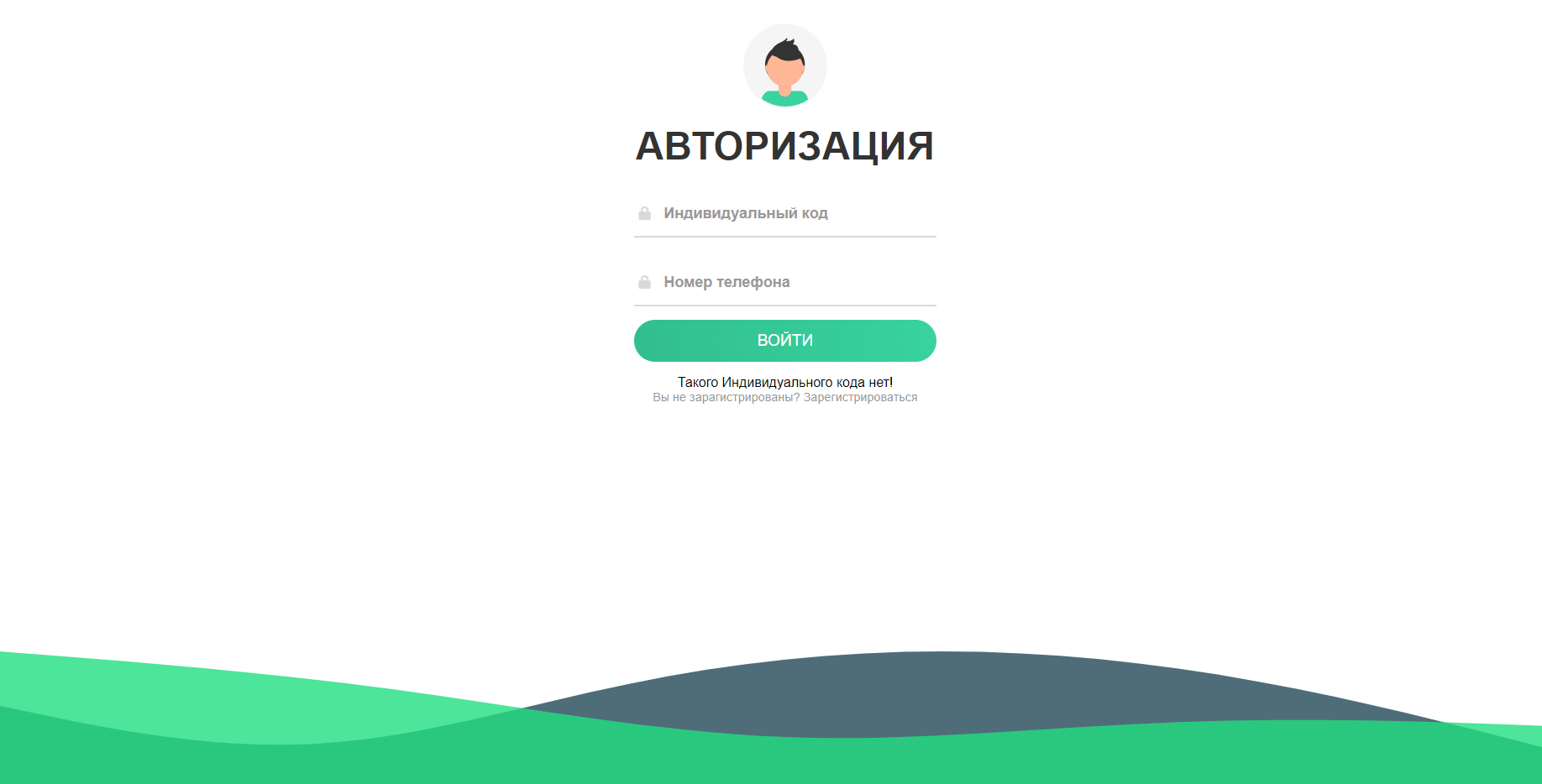


Рисунок Б.5 – Оповещение при неудачной авторизации

Данные пользователя можно редактировать, введя актуальные данные в поля формы и нажав на кнопку «Сохранить». Как представлено на рисунке Б.6

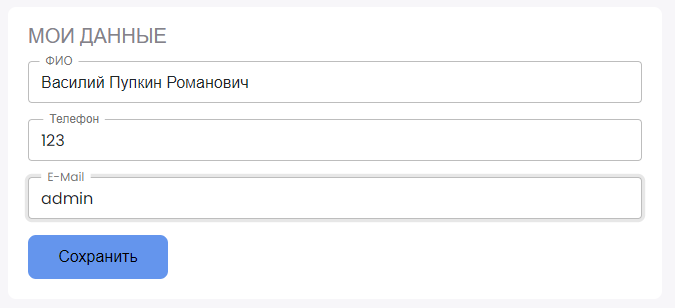


Рисунок Б.6 – Форма изменения данных пользователя