**Поволжский Государственный Университет Телекоммуникаций и Информатики**Кафедра «Программная инженерия»

|  |  |
| --- | --- |
| Сдана на проверку  «\_\_» \_\_\_\_2024 г. | Допустить к защите  «\_\_» \_\_\_\_\_ 2024 г.  Защищена с оценкой  «\_\_\_» \_\_\_\_\_\_\_\_ 2024 г. |

**КУРСОВАЯ РАБОТА**По дисциплине: «Прикладное программирование»  
На тему: «Разработка клиент-серверного web-приложения по поиску команды в Startup»

|  |  |
| --- | --- |
| Студент группы РПИС-12 \_\_\_\_\_\_ Правдин Денис  (роспись) (ФИО)  Руководитель \_\_\_\_\_\_\_\_ к.т.н., доц. Ахметшина Э. Г.  (роспись) (ФИО) | 0 210418 0  (№ зачетной книжки) |

Самара 2024

**Рецензия**

Содержание

[1 Описание предметной области. Актуальность 4](#_Toc156243022)

[2 Описание программы. Общие сведения 5](#_Toc156243023)

[3 Функциональное назначение 5](#_Toc156243024)

[4 Логическая модель базы данных 7](#_Toc156243025)

[5 Физическая модель базы данных 9](#_Toc156243026)

[6 Диаграмма классов 12](#_Toc156243027)

[7 Диаграмма компонентов 14](#_Toc156243028)

[8 Диаграмма вариантов использования 15](#_Toc156243029)

[9 Диаграмма последовательности 16](#_Toc156243030)

[10 Демонстрация работы приложения 16](#_Toc156243031)

[Список используемых источников 18](#_Toc156243032)

[Приложение А – Листинг программного кода серверной части 19](#_Toc156243033)

[Приложение Б – Листинг программного кода клиентская части 30](#_Toc156243034)

# **1 Описание предметной области. Актуальность**

В современном мире успешный стартап немыслим без сильной и сплоченной команды. Однако найти подходящих специалистов, которые разделяют ваше видение и готовы вложить свои навыки и усилия в развитие проекта, может быть сложной задачей.

Именно поэтому сайт по поиску команды для стартапа является актуальным и необходимым инструментом для предпринимателей и основателей новых бизнес-проектов. Сайт предлагает уникальную платформу, которая поможет пользователю найти и привлечь талантливых профессионалов, готовых работать вместе с пользователем над развитием вашего стартапа.

Вот где актуальность сайта по поиску команд для стартапа становится очевидной. Такой сайт помогает пользователю найти подходящих кандидатов для своего стартапа.

Такой сайт может предложить расширенную базу данных специалистов, с различным опытом работы и навыками, чтобы пользователи могли выбрать наиболее подходящих кандидатов для вашего проекта.

Еще одно преимущество такого сайта - конфиденциальность и безопасность. Пользователь может быть уверен, что данные и процессы на сайте будут защищены.

Таким образом, актуальность сайта по поиску команды для стартапа очевидна, поскольку предприниматели ищут удобные инструменты для нахождения талантливых специалистов, которые помогут им воплотить свои идеи и достичь успеха со своими стартапами.

# **2 Описание программы. Общие сведения**

Приложение – веб-ресурс. Установка не требуется.

Необходимые требования: доступ к интернету.

Язык программирования: Java, JavaScript.

Фреймворки: ReactJS, Spring Boot.

Платформа: Браузер.

Среда разработки: Visual Studio Code, IntelliJ IDEA.

Объем проекта: 69,3 МБ (вместе с исходным кодом).

Исходный код (классы и код форм): 1819 строк.

# **3 Функциональное назначение**

Функциональное назначение сайта по поиску команды для стартапа заключается в том, чтобы помочь предпринимателям искать и нанимать специалистов для своих проектов. Основные функции такого сайта включают:

1. Поиск:

Сайт предоставляет возможность искать потенциальных участников команды по определенным критериям, таким как область экспертизы, технологические навыки, опыт работы.

Для поиска специалистов или интересного стартапа на сайте есть отдельные страницы, где выводятся определенная информация по ним.

1. Профили команды:

Каждая команда имеет свой профиль на сайте, где указаны навыки, опыт работы, образование и портфолио. Это позволяет предпринимателю оценить профессиональный потенциал и характеристики кандидата.

1. Профили Стартапов:

Для каждого нового созданного стартапа была сделана отдельная страница с отображением профиля и информацией о стартапе, и откликов на нее. Также пользователь, который создал его, может изменять данные на ней.

1. Отклики команд на стартап:

Любая команда может откликнуться на любой понравившийся стартап и также удалить своей отклик. Пользователь, который создал стартап, может просматривать профили откликнувшихся команда и принимать решение о сотрудничестве с ними, с помощью соответствующих кнопок.

Таким образом, функциональное назначение сайта по поиску команды для стартапа заключается в предоставлении инструментов для эффективного подбора и найма специалистов, основываясь на предъявляемых требованиях и ожиданиях предпринимателя.

# **4 Логическая модель базы данных**

Логическая модель базы данных показывает структуру данных в базе и взаимосвязь между ними без учета физической реализации. Она описывает сущности, атрибуты и связи между ними.

В логической модели базы данных сайта есть 5 сущностей: Стартапы, Пользователи, Команды, Члены Команд, Связь команды к стартапу. Также подробное описание сущностей со своими атрибутами содержится в таблице 4.1

Таблица 4.1

Описание сущностей логической модели данных

|  |  |
| --- | --- |
| **Сущность** | **Атрибуты** |
| Cтартапы | Id\_Cтартапа, id\_Cоздателя, Название, О стартапе |
| Пользователи | Id\_Пользователя, Фамилия, Имя, Отчество, Контактные данные, NickName/Login, Password, О себе |
| Члены Команды | Id\_связи, id\_Команды, Фамилия, Имя, Роль |
| Команды | Id\_Команды, id\_Создателя, Контактные данные, Название команды, Ссылка на портфолио, Опыт Команды, О команде |
| Связь команды к стартапу | Id\_Связи, id\_Стартапа, id\_Команды, Статус связи. |

Описание связей между сущностями:

1. Сущность “Стартапы” связана со сущностью “Пользователь” через поля id\_Создателя-id\_Пользователя, связь – один ко многим;
2. Сущность “Стартапы” связана со сущностью “Связь Команды к стартапу” через поля id\_Стартапа, связь – один ко многим;
3. Сущность “Команды” связана со сущностью “Связь Команды к стартапу” через поля id\_Команды, связь – один ко многим;
4. Сущность “Команды” связана со сущностью “Пользователь” через поля id\_Создателя-id\_Пользователя, связь – один к одному;
5. Сущность “Команды” связана со сущностью “Члены Команды” через поля id\_Команлы, связь – один ко многим;

Логическая модель базы данных представлена на рис. 4.1.

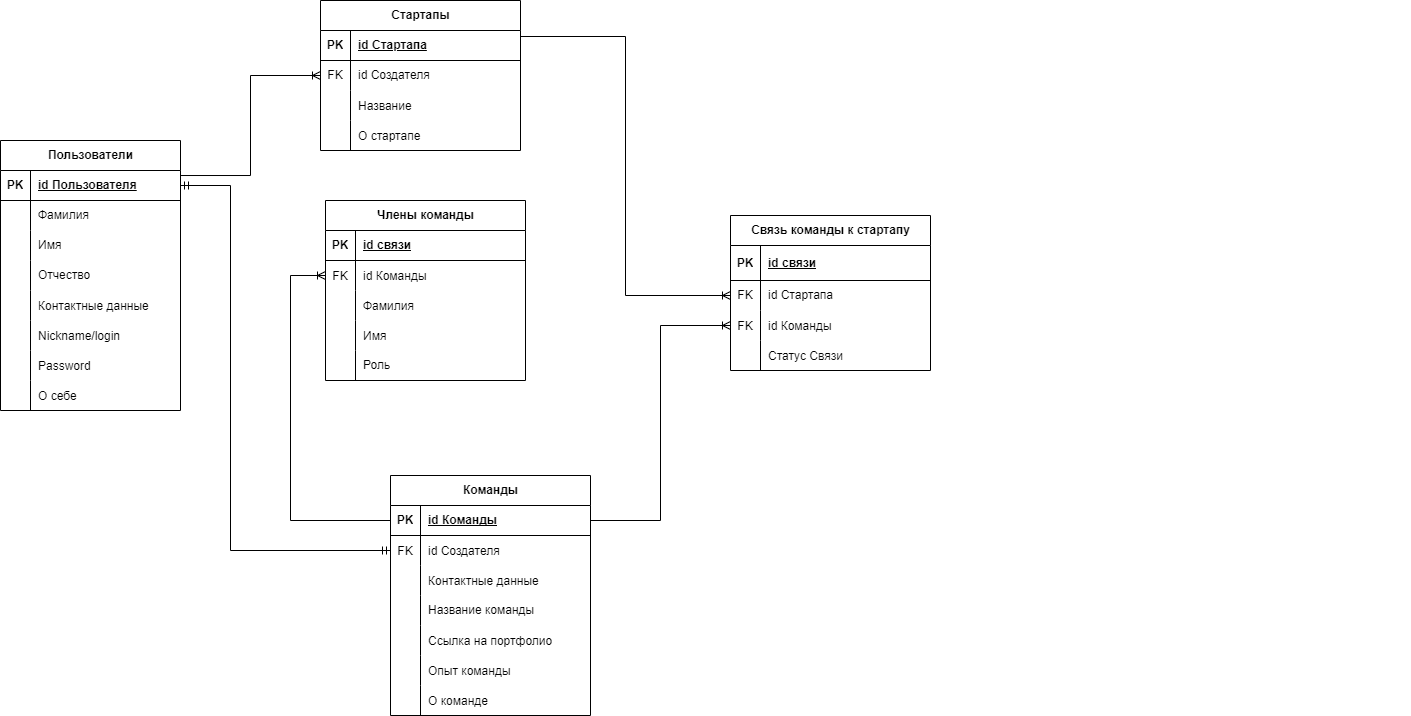


Рис 4.1 - Логическая модель базы данных

# **5 Физическая модель базы данных**

Физическая модель базы данных описывает физическое представление данных в базе данных. Она учитывает детали реализации базы данных, такие как типы данных, структура хранения, ограничения целостности, индексы и прочие факторы, которые могут влиять на производительность и эффективность работы с базой данных.

Таблица 5.1

Физическая модель базы данных

|  |  |
| --- | --- |
| **Сущность** | **Атрибуты** |
| Start | Id\_creatst, id\_Start, name\_start, about\_start |
| Users | Id\_User, user\_surname, user\_name , user\_patronic, user\_messeger, user\_login , user\_password, user\_about |
| BundleUC | Id\_bundle, id\_Command, surname\_part, name\_part, role\_part |
| Command | Id\_Command-, id\_creater, contact\_Command, name\_Command, link\_Command, Exp\_Command, About\_Command |
| Связь команды к стартапу | Id\_bundle, id\_Start, id\_Command, Status\_Bundle. |

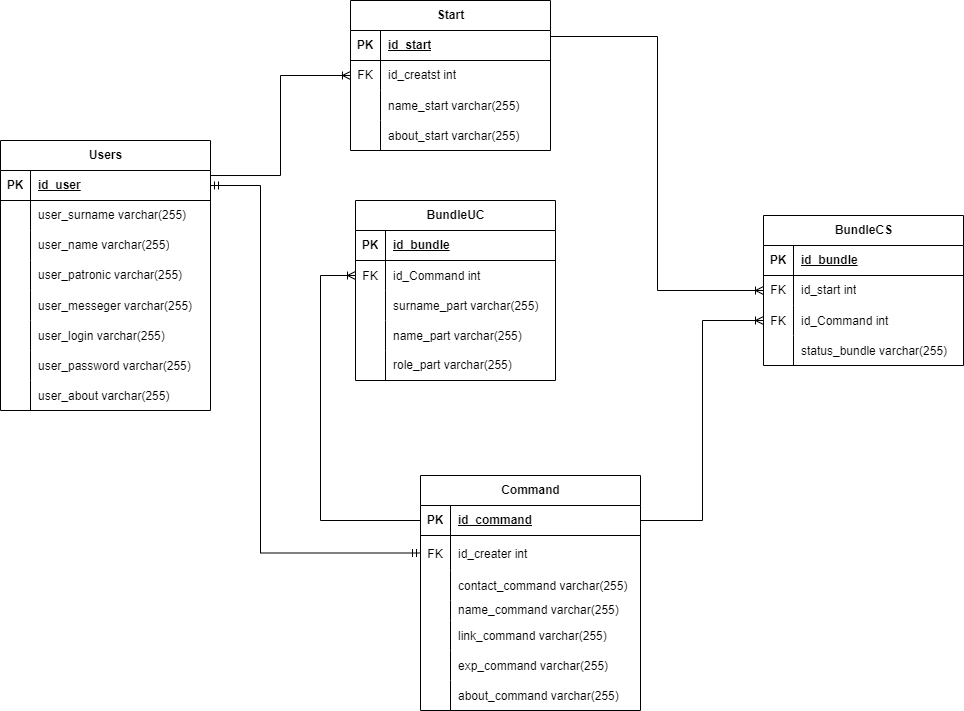


Рис.5.1 – Физическая модель базы данных

Вся физическая модель базы данных была перенесена в СУБД PostgreSQL (рис.5.2-5.6).



Рис.5.2 – Сущность “Users”

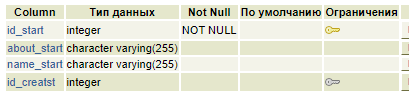


Рис.5.3 – Сущность “Start”

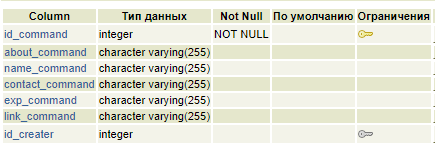


Рис.5.3 – Сущность “Command”

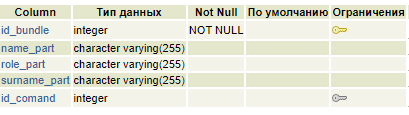


Рис.5.4 – Сущность “BundleUC”

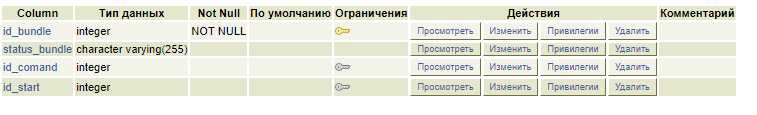


Рис.5.5 – Сущность “BundleCS”



Рис.5.6 – Полная База данных

# **6 Диаграмма классов**

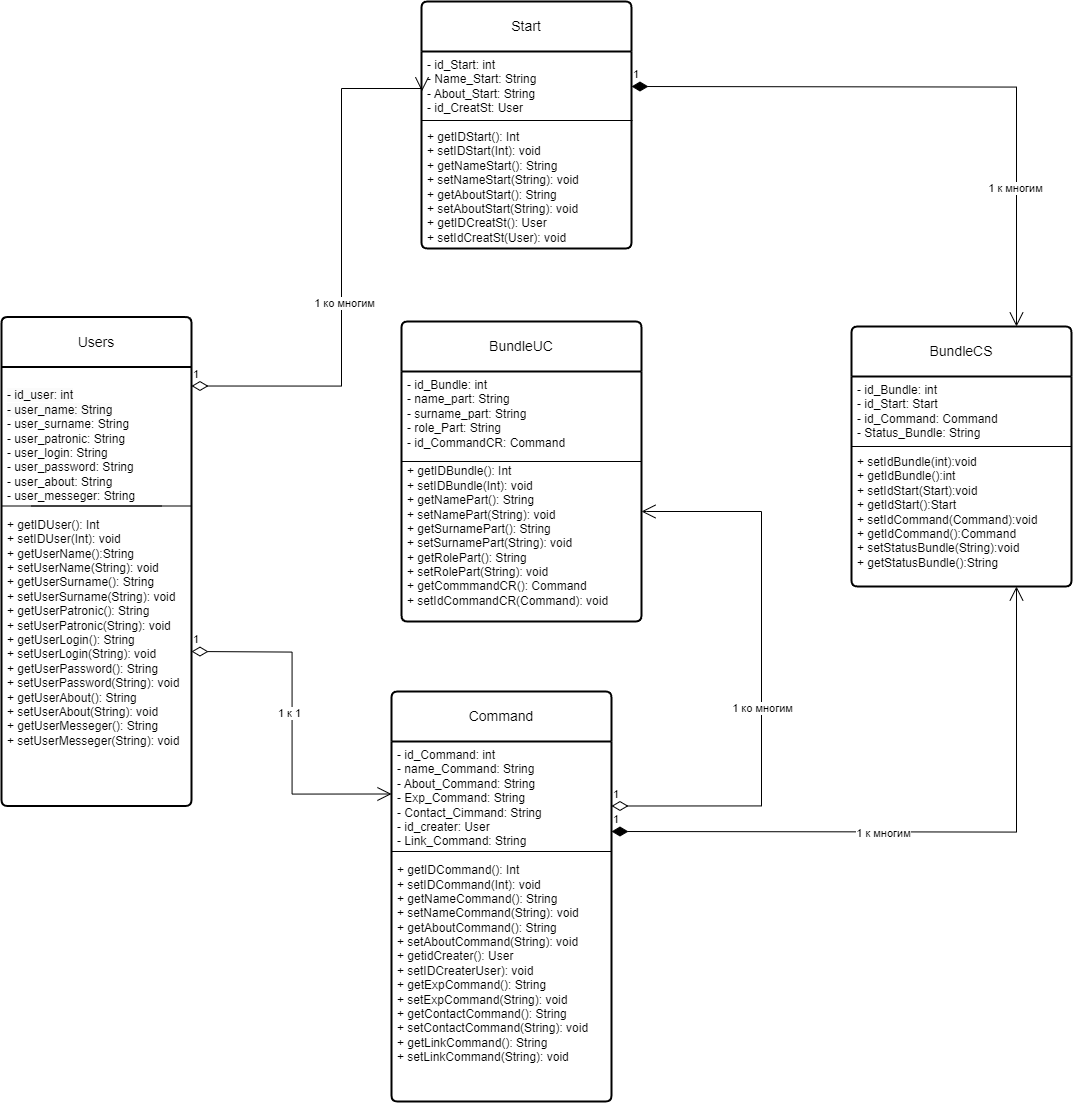


Рис. 6.1 – Диаграмма классов

# **7 Диаграмма компонентов**

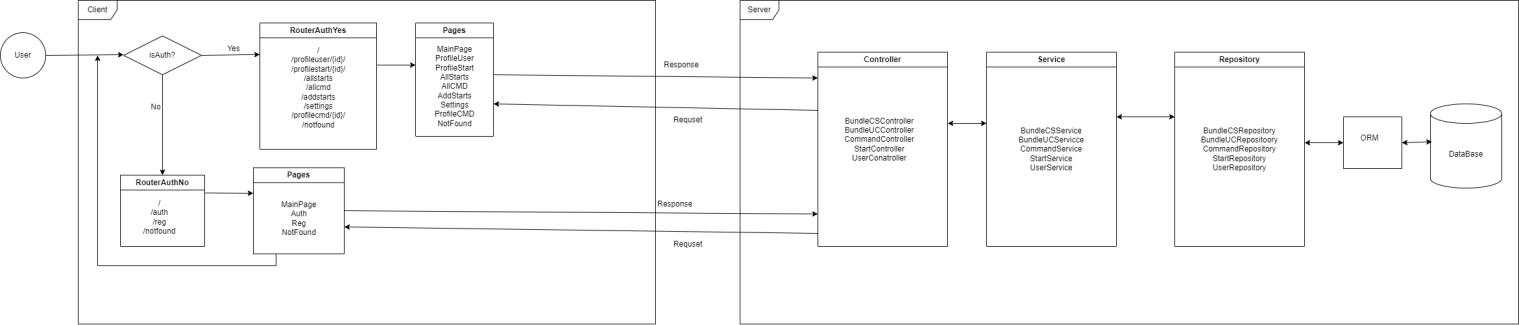


Рис. 7.1 – Диаграмма компонентов

# **8 Диаграмма вариантов использования**

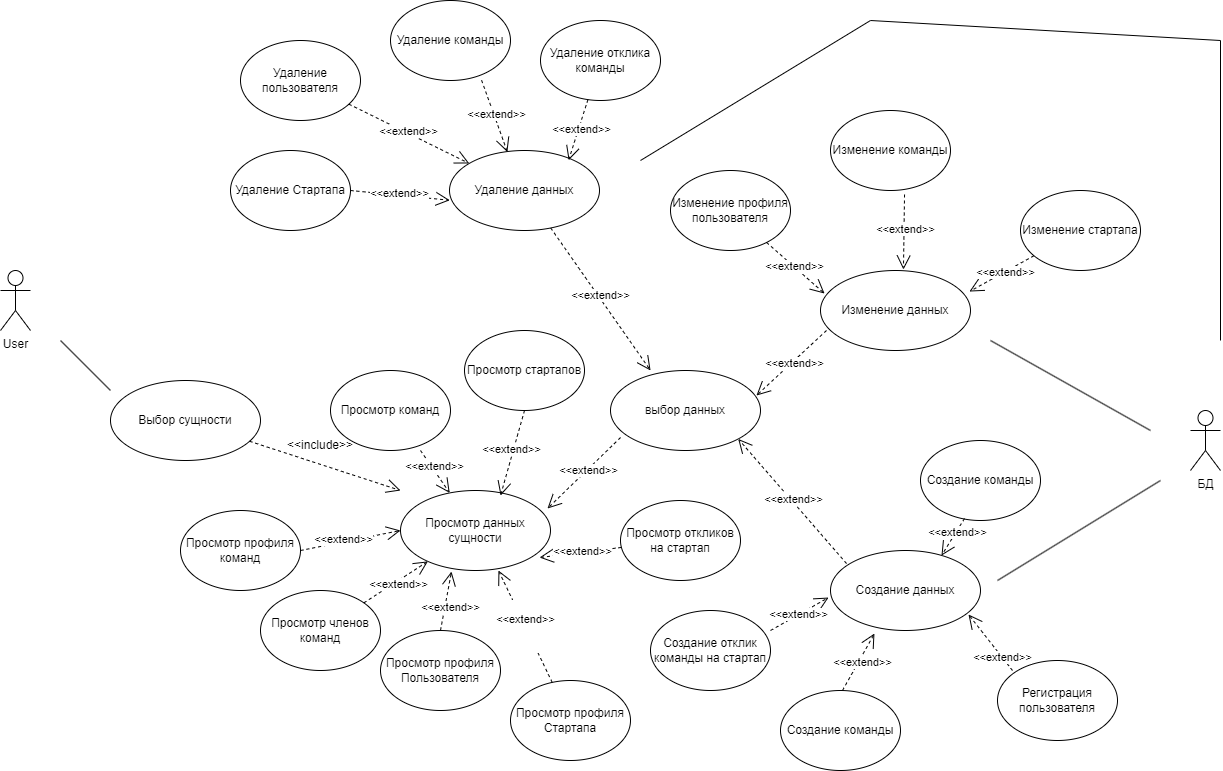


Рис. 8.1 – Диаграмма вариантов использования

# **9 Диаграмма последовательности**

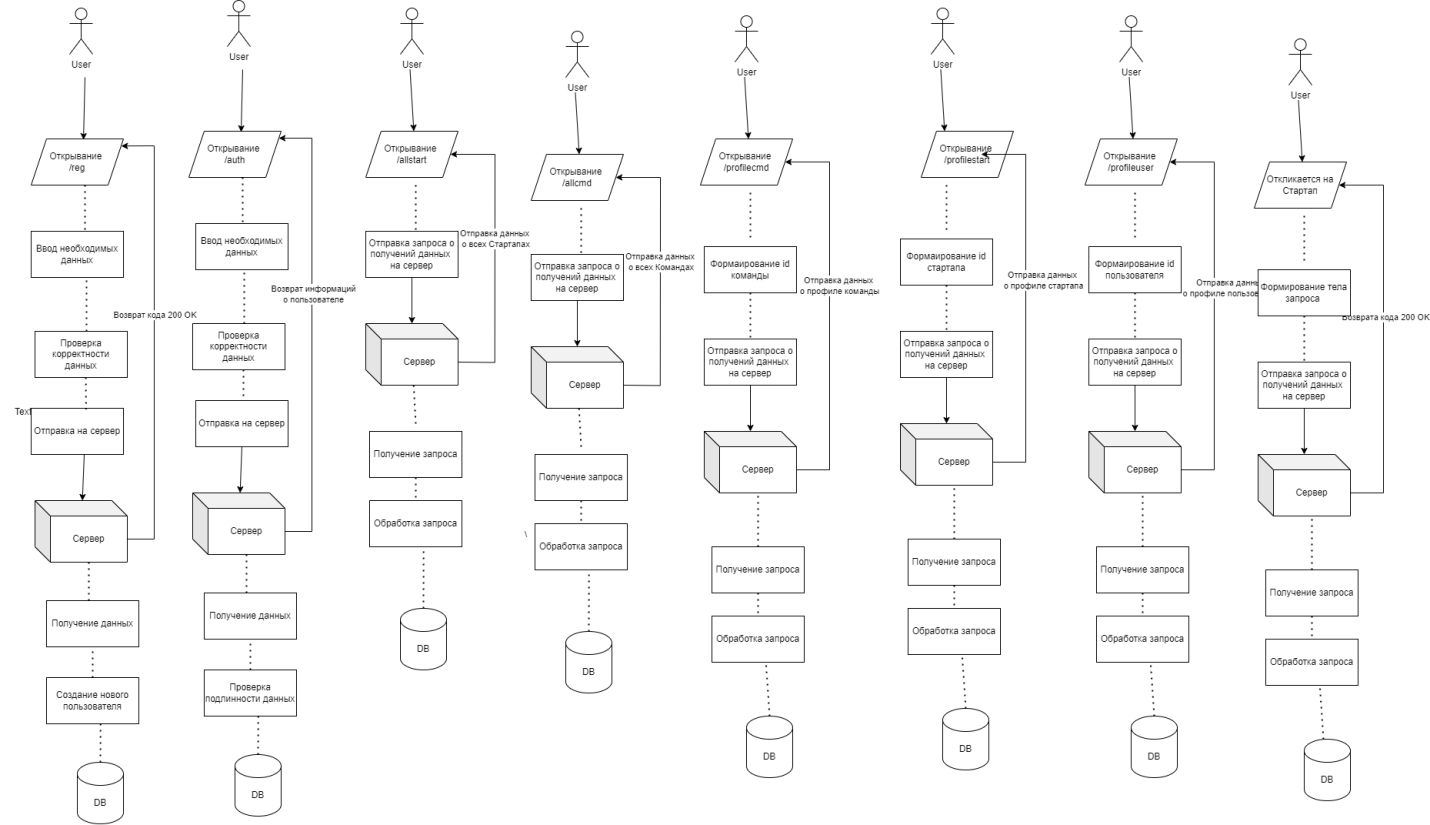


Рис. 9.1 – Диаграмма вариантов использования

# **10 Демонстрация работы приложения**

Для захода на сайт необходимо перейти по домену сайта или по IP-адресу, на котором был развернут.

Для неавторизованных пользователей будут доступны всего лишь 3 страницы: Главная страница (рис. 10.1), Регистрация(рис. 10.2), Страница Авторизаций (рис. 10.3).

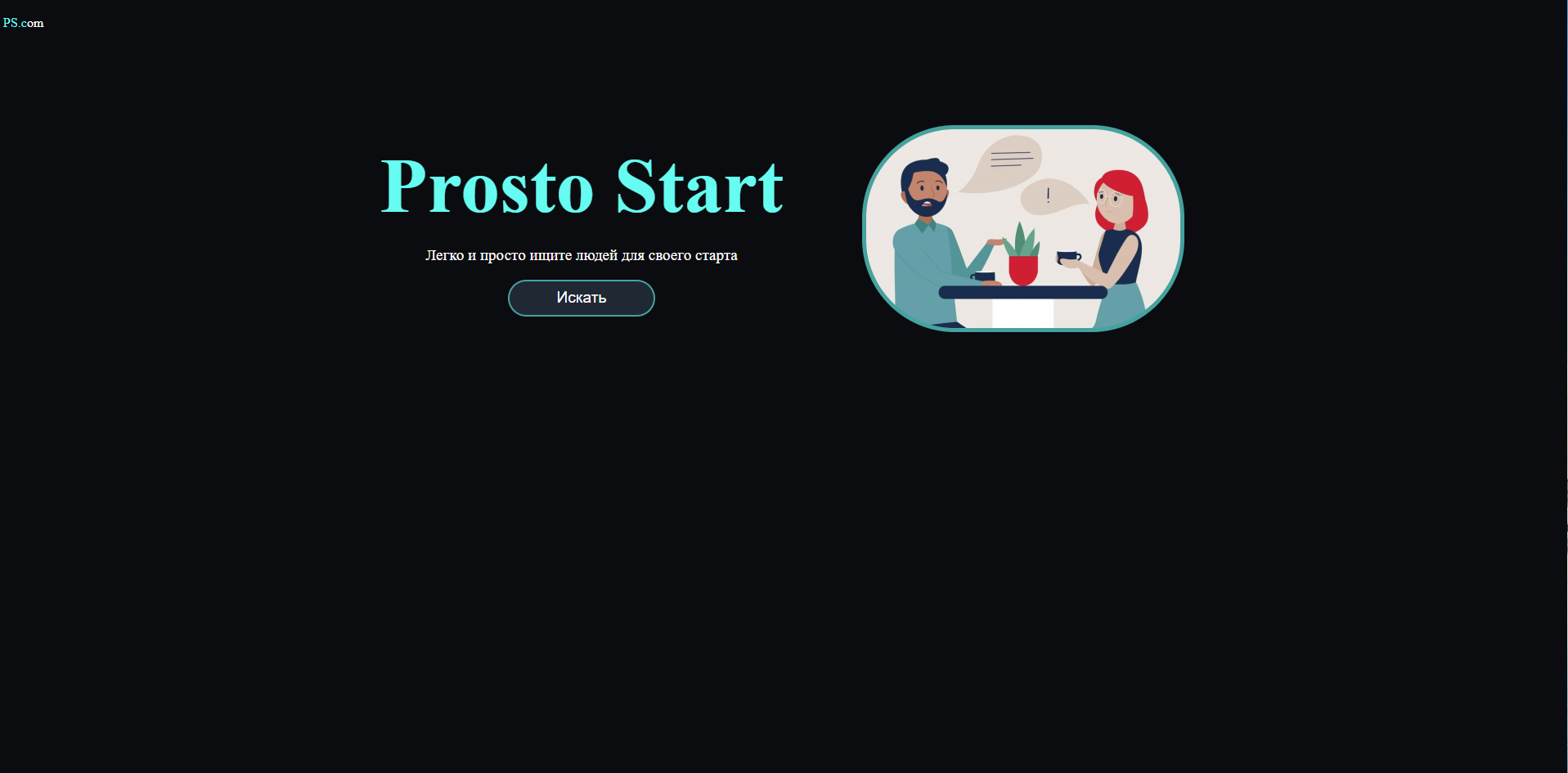


Рис.10.1 – Главная страница

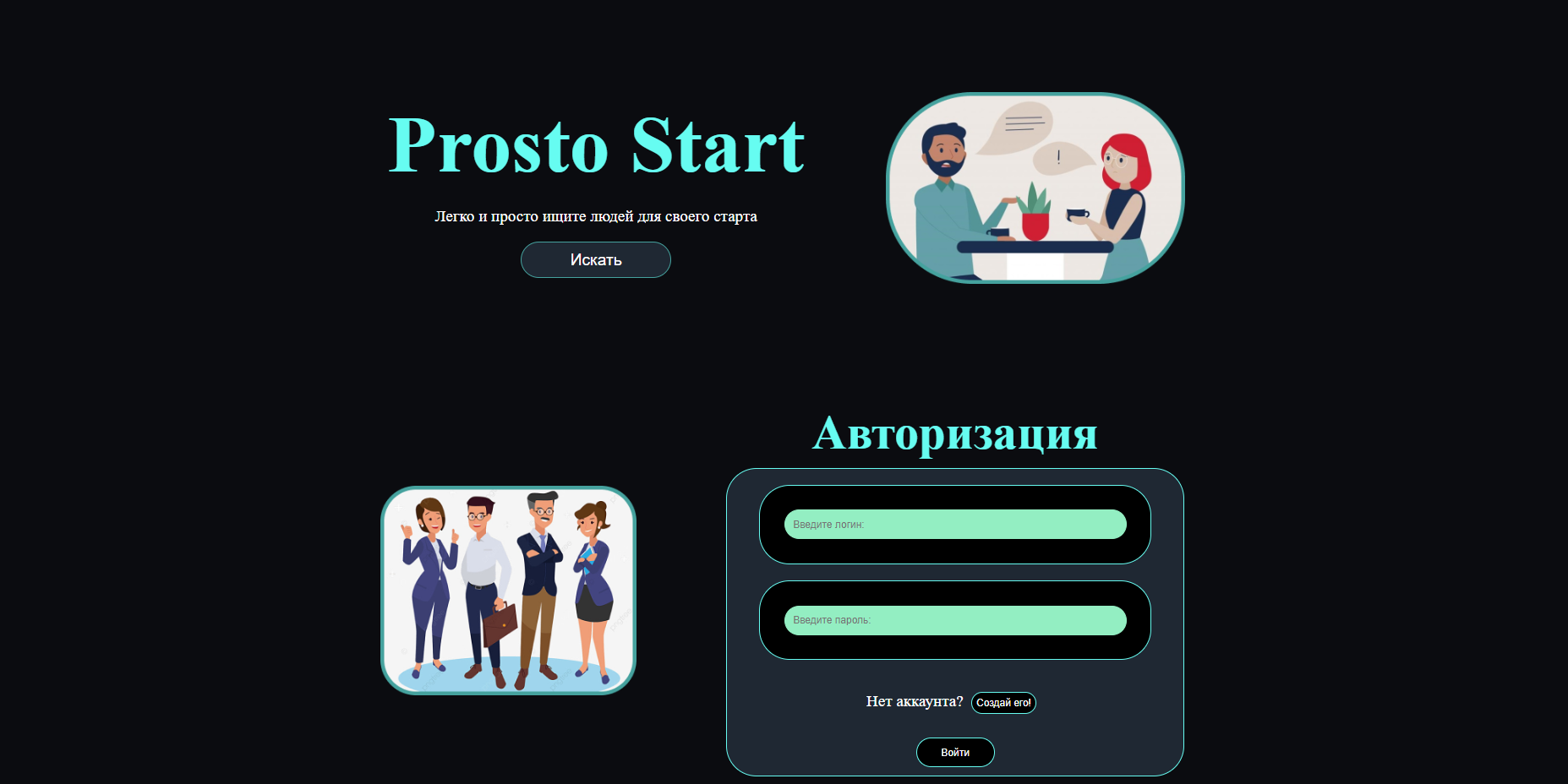


Рис.10.2 – Страница авторизации

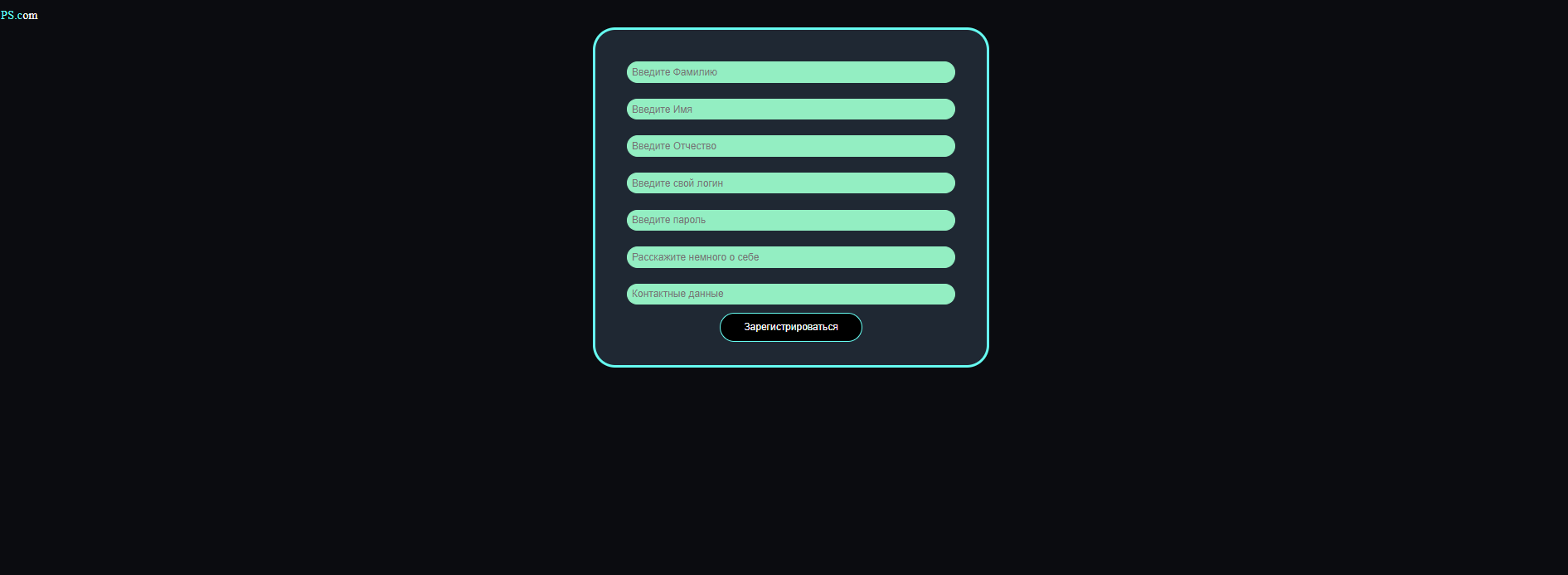


Рис.10.3 – Страница Регистрации

После авторизаций пользователю станут доступны все остальные страницы (рис.10.4 – 10.), где он может опробовать весь основной функционал сайта.

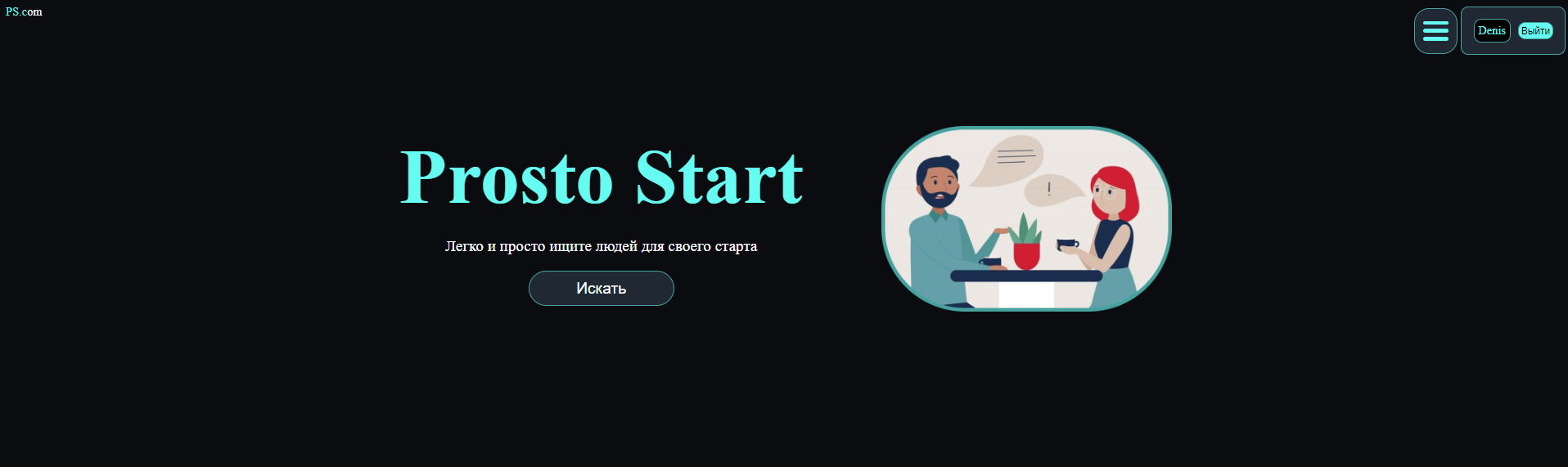


Рис.10.4 – Главная страница



Рис.10.5 – Страница со всеми стартапами

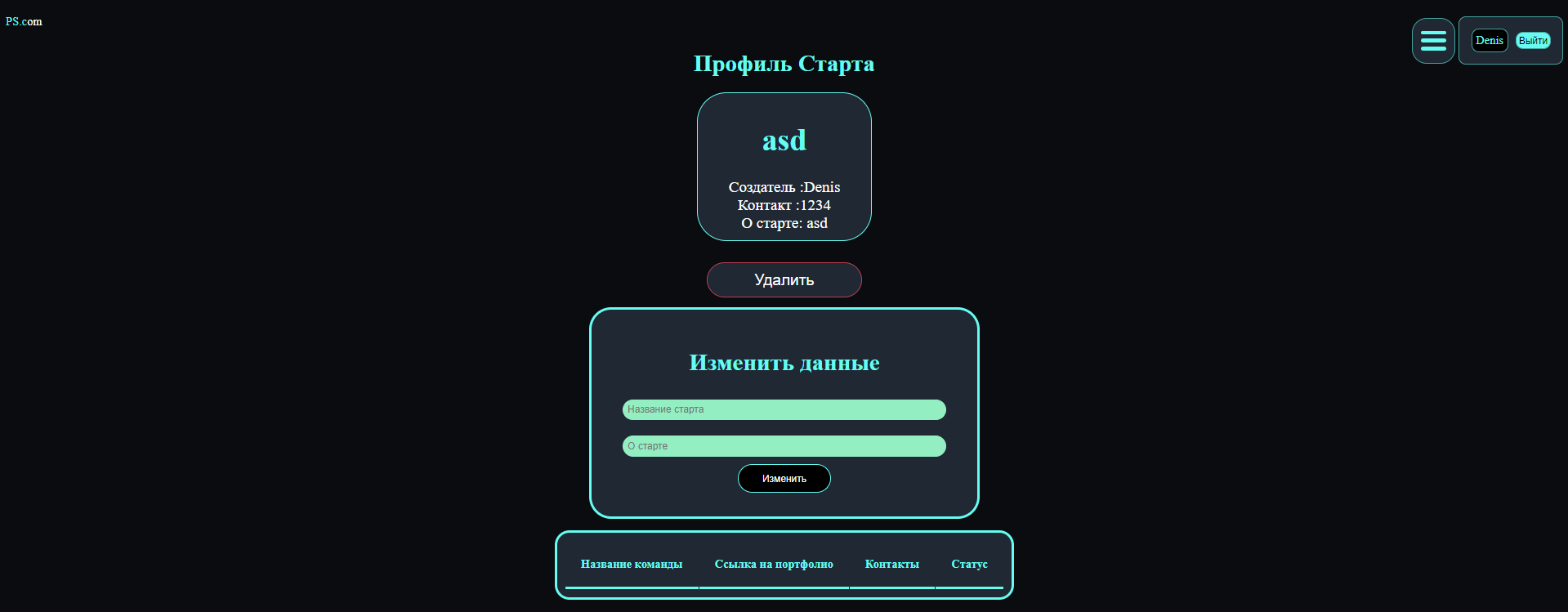


Рис.10.6 – Профиль стартапа

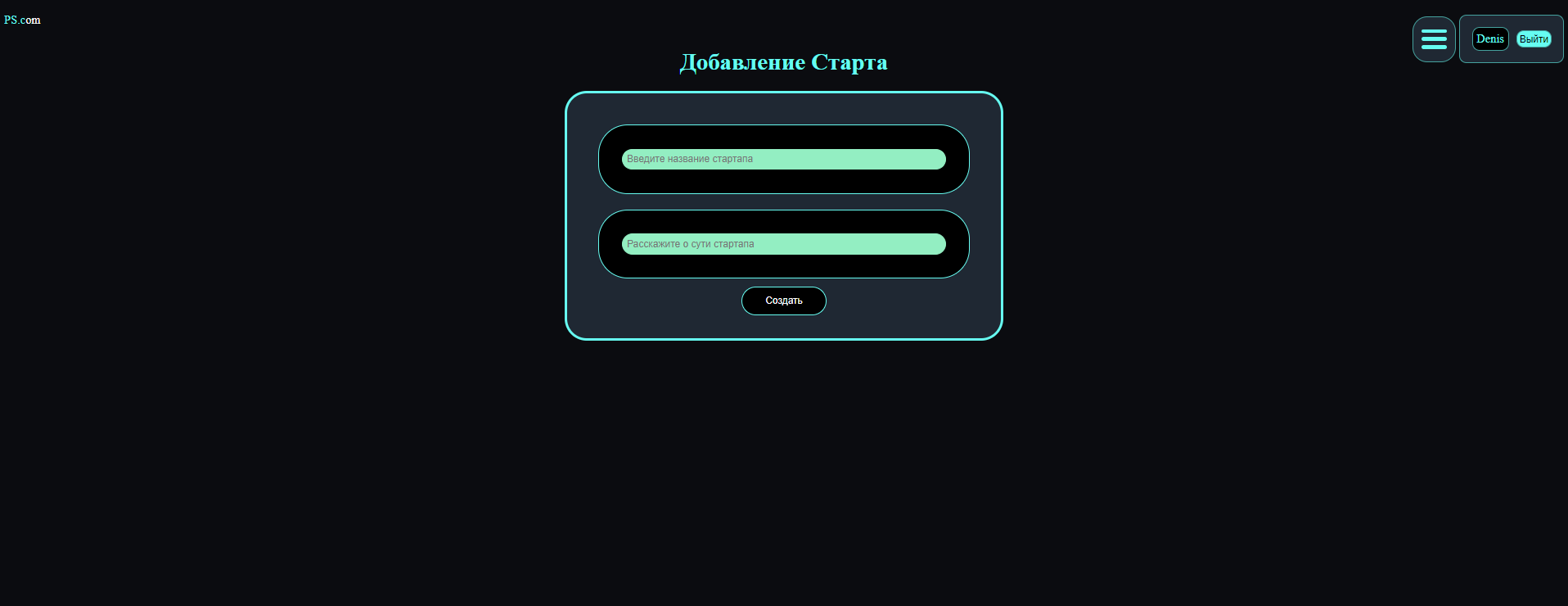


Рис.10.7 – Добавление стартапа

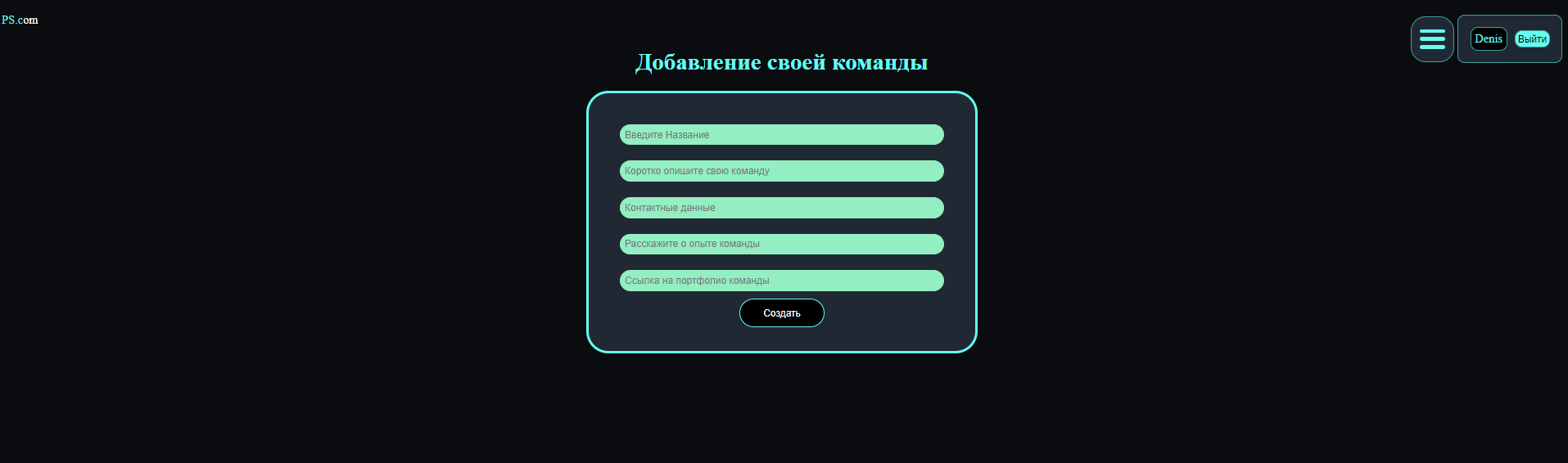


Рис.10.8 – Добавление команды

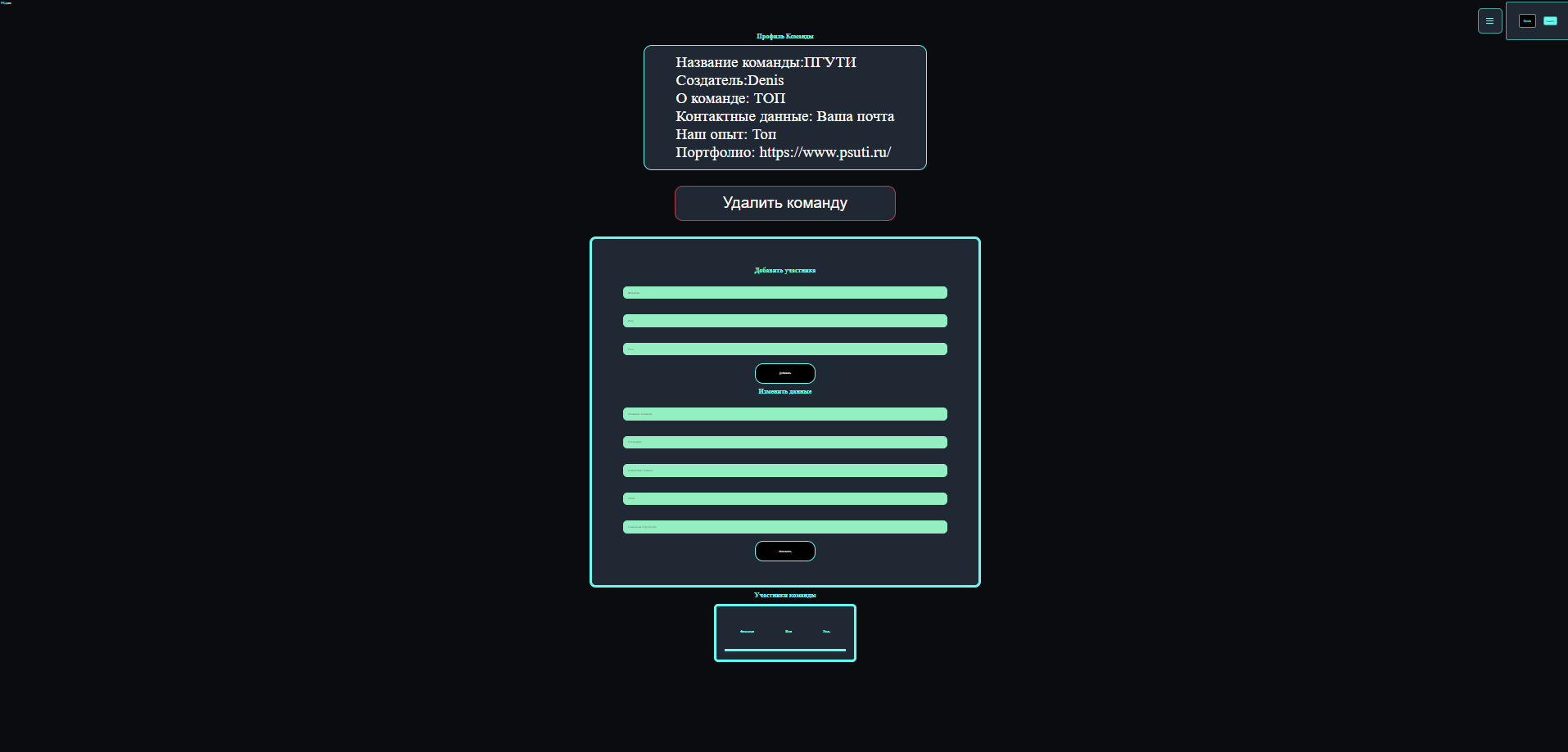


Рис.10.9 – Профиль команды

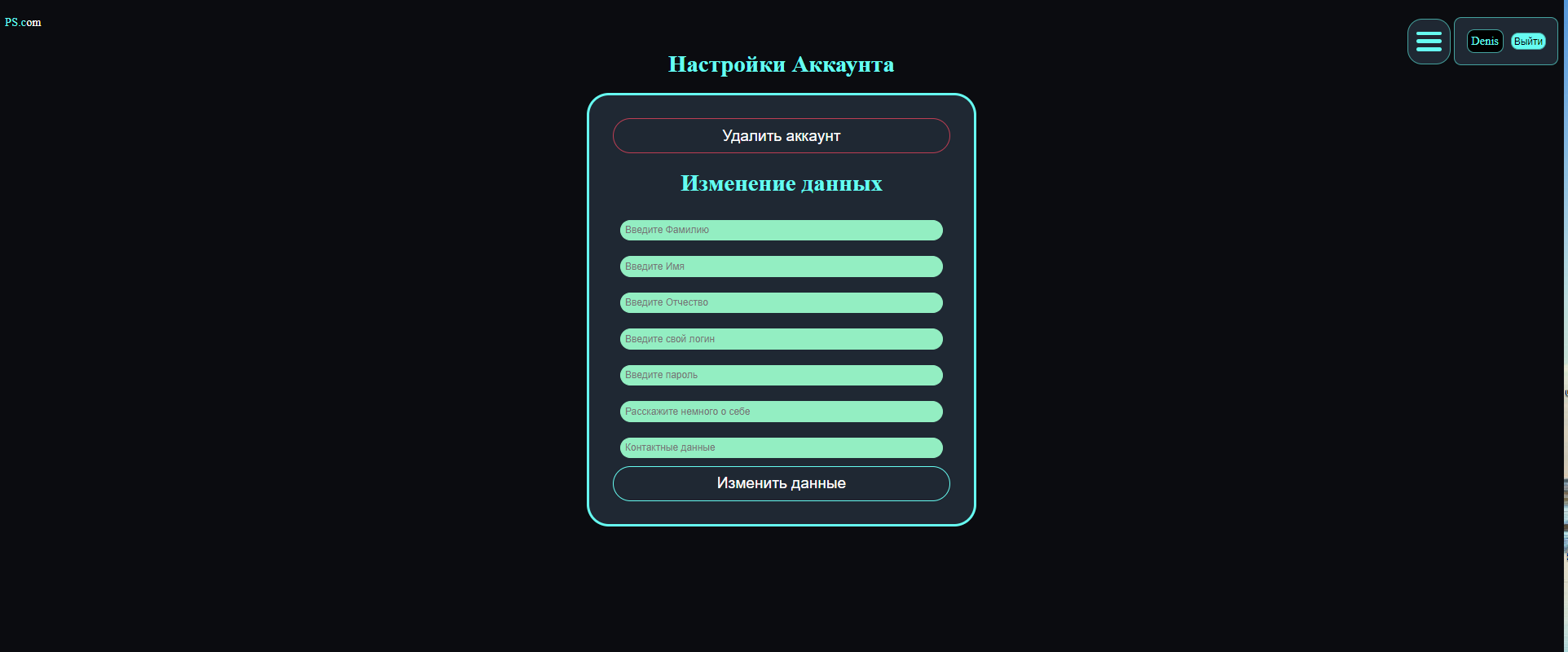


Рис.10.10 – Настройки Аккаунта

# **Список используемых источников**

1. Начало работы / [Электронный ресурс]. - Электронные данные. - Режим доступа: https://ru.legacy.reactjs.org/docs/getting-started.html
2. Spring Quickstart Guide / [Электронный ресурс]. - Электронные данные. - Режим доступа: https://spring.io/quickstart
3. Tutorial / [Электронный ресурс]. - Электронные данные. - Режим доступа: https://reactrouter.com/en/main/start/tutorial
4. Getting Spring Security / [Электронный ресурс]. - Электронные данные. - Режим доступа: https://docs.spring.io/spring-security/reference/getting-spring-security.html
5. Getting Started with Redux / [Электронный ресурс]. - Электронные данные. - Режим доступа: https://redux.js.org/introduction/getting-started
6. А. Хрусталев, А. Кириченко "HTMLS + CSS3. Основы современного WEB-дизайна", Наука и Техника, 2021, 352 стр

# **Приложение А – Листинг программного кода серверной части**

@RestController  
@CrossOrigin  
@RequestMapping("/bundlecs")  
public class BundleCSController {  
  
 @Autowired  
 private BundleCSService bundleCSService;  
 @PostMapping("/create")  
 public ResponseEntity<HttpStatus> createBundle(@RequestBody BundleCS bundleCS){  
 bundleCSService.save(bundleCS);  
 return ResponseEntity.ok(HttpStatus.OK);  
 }  
 @GetMapping("/all")  
 public List<BundleCS> getAll()  
 {  
 return bundleCSService.getall();  
 }  
 @GetMapping("/change")  
 public ResponseEntity<HttpStatus> changeBudle(@RequestHeader String status\_bundle, @RequestParam long id\_bundlecs)  
 {  
 bundleCSService.change(status\_bundle, id\_bundlecs);  
 return ResponseEntity.ok(HttpStatus.OK);  
 }  
 @GetMapping("/listbundle")  
 public List<BundleCS> getidStart(@RequestHeader long id\_start){  
 return bundleCSService.getIDStart(id\_start);  
 }  
 @GetMapping("/delete")  
 public ResponseEntity<HttpStatus> deleteBundle(@RequestHeader long id\_bundle){  
 bundleCSService.delete(id\_bundle);  
 return ResponseEntity.ok(HttpStatus.OK);  
 }  
}

@RestController  
@CrossOrigin  
@RequestMapping("/bundleuc")  
public class BundleUCController {  
  
 @Autowired  
 private BundleUCService bundleUCService;  
 @Autowired  
 private CommandService commandService;  
 @PostMapping("/create")  
 public ResponseEntity<HttpStatus> createBundleUC(@RequestBody BundleUC bundleuc){  
 bundleUCService.create(bundleuc);  
 return ResponseEntity.ok(HttpStatus.OK);  
 }  
 @GetMapping("/all")  
 public java.util.List<BundleUC> getAll(){  
 return bundleUCService.getAll();  
 }  
 @GetMapping("/delete")  
 public ResponseEntity<HttpStatus> deleteUser(@RequestParam long id\_bundleuc){  
 bundleUCService.deletebundle(id\_bundleuc);  
 return ResponseEntity.ok(HttpStatus.OK);  
 }  
}

@RestController  
@CrossOrigin  
@RequestMapping("/cmd")  
public class CommandController {  
  
 @Autowired  
 private CommandService commandService;  
 UserService userService;  
 UserRepository userRepository;  
 @PostMapping("/create")  
 public ResponseEntity<HttpStatus> CMDcreate(@RequestBody Command command){  
 commandService.save(command);  
 return ResponseEntity.ok(HttpStatus.OK);  
 }  
 @GetMapping("/all")  
 public List<Command> getAllCMD(){  
 return commandService.getall();  
 }  
 @GetMapping("/prof")  
 public Optional<Command> getOneCMD(@RequestHeader long id\_CMD) {return commandService.getCommand(id\_CMD);}  
 @GetMapping("/delete")  
 public ResponseEntity<HttpStatus> deleteUser(@RequestHeader long id\_command){  
 commandService.deleteComm(id\_command);  
 return ResponseEntity.ok(HttpStatus.OK);  
 }  
 @GetMapping("/change")  
 public ResponseEntity changeUser( @RequestHeader String name\_Comand, @RequestHeader String About\_Comand, @RequestHeader String Exp\_Comand,@RequestHeader String Contact\_Comand, @RequestHeader String Link\_Comand, @RequestParam long id\_Comand){  
 commandService.changeComm(name\_Comand, About\_Comand, Exp\_Comand, Contact\_Comand, Link\_Comand, id\_Comand);  
 return ResponseEntity.ok(HttpStatus.OK);  
 }  
 @GetMapping("/commuser")  
 public List<Command> getCMD(@RequestHeader long id\_CMD) {return commandService.getCommID(id\_CMD);}  
}

@RestController  
@CrossOrigin  
@RequestMapping("/start")  
public class StartController {  
  
 @Autowired  
 private StartService startService;  
 @PostMapping("/create")  
 public ResponseEntity<HttpStatus> createStart(@RequestBody Start start){  
 startService.save(start);  
 return ResponseEntity.ok(HttpStatus.OK);  
 }  
 @GetMapping("/all")  
 public List<Start> getAllStart(){  
 return startService.getall();  
 }  
 @GetMapping("/profile")  
 public Optional<Start> profStart(@RequestHeader long id\_Start){  
 return startService.getStart(id\_Start);  
 }  
 @GetMapping("/delete")  
 public ResponseEntity<HttpStatus> deleteStart(@RequestHeader long id\_Start){  
 startService.deleteStart(id\_Start);  
 return ResponseEntity.ok(HttpStatus.OK);  
 }  
 @GetMapping("/change")  
 public ResponseEntity<HttpStatus> changeStart(@RequestHeader String Name\_Start, @RequestHeader String About\_Start, @RequestParam long id\_Start){  
 startService.changeStart(Name\_Start, About\_Start, id\_Start);  
 return ResponseEntity.ok(HttpStatus.OK);  
 }  
 @GetMapping("/getuserstart")  
 public List<Start> getUserStart(@RequestHeader long id\_user){  
 return startService.findByidStart(id\_user);  
 }  
  
}

@RestController  
@CrossOrigin  
@RequestMapping("/user")  
public class UserController {  
 @Autowired  
 private UserService userService;  
 @GetMapping("/all")  
 public List<User> getAlluser(){  
 return userService.getall();  
 }  
 @PostMapping("/reg")  
 public ResponseEntity regUser(@RequestBody User user){  
 userService.save(user);  
 return ResponseEntity.ok(HttpStatus.OK);  
 }  
 @GetMapping("/auth")  
 public List<User> authUser(@RequestHeader String user\_login, @RequestHeader String user\_password){  
 return userService.auth(user\_login,user\_password);  
  
 }  
 @GetMapping("/profile")  
 public Optional<User> authUser(@RequestHeader long id\_user){  
 return userService.getUser(id\_user);  
  
 }  
 @GetMapping("/change")  
 public ResponseEntity changeUser( @RequestHeader String user\_name, @RequestHeader String user\_surname, @RequestHeader String user\_patronic,@RequestHeader String user\_login, @RequestHeader String user\_password, @RequestHeader String user\_about,@RequestHeader String user\_messager, @RequestParam long id\_user){  
 userService.changeUser( user\_name, user\_surname, user\_patronic,id\_user,user\_messager, user\_password,user\_about,user\_login);  
 return ResponseEntity.ok(HttpStatus.OK);  
 }  
 @GetMapping("/delete")  
 public ResponseEntity<HttpStatus> deleteUser(@RequestHeader long id\_user){  
 userService.deleteUser(id\_user);  
 return ResponseEntity.ok(HttpStatus.OK);  
 }  
}

@Getter  
@Setter  
@Entity  
@Table(name = "BundleCS")  
@NoArgsConstructor  
@AllArgsConstructor  
public class BundleCS implements Serializable {  
 @Id  
 @GeneratedValue(strategy = GenerationType.AUTO)  
 @Column(name="id\_Bundle")  
 private int id\_Bundle;  
  
 @ManyToOne  
 @JoinColumn(name="id\_Start")  
 private Start id\_Start;  
  
 @ManyToOne  
 @JoinColumn(name="id\_Comand")  
 private Command id\_Command;  
  
 @Column(name="Status\_Bundle")  
 private String Status\_Bundle;  
}

@Getter  
@Setter  
@Entity  
@NoArgsConstructor  
@Table(name="BundleUC")  
@AllArgsConstructor  
public class BundleUC implements Serializable {  
 @Id  
 @GeneratedValue(strategy = GenerationType.AUTO)  
 @Column(name="id\_Bundle")  
 private int id\_Bundle;  
 @Column(name="name\_Part")  
 private String name\_Part;  
 @Column(name="surname\_Part")  
 private String surname\_Part;  
 @Column(name="role\_Part")  
 private String role\_Part;  
 @ManyToOne  
 @JoinColumn(name="id\_Comand")  
 private Command id\_CommandCR;  
}

@Getter  
@Setter  
@Entity  
@NoArgsConstructor  
@AllArgsConstructor  
@Table(name="Command")  
public class Command implements Serializable {  
 @Id  
 @GeneratedValue(strategy = GenerationType.AUTO)  
 @Column(name="id\_command")  
 private int id\_Comand;  
 @Column(name = "name\_command")  
 private String name\_Comand;  
 @Column(name="about\_command")  
 private String About\_Comand;  
 @Column(name="exp\_command")  
 private String Exp\_Comand;  
 @Column(name="contact\_command")  
 private String Contact\_Comand;  
 @OneToOne  
 @JoinColumn(name="id\_creater")  
 private User id\_creater;  
 @Column(name="link\_command")  
 private String Link\_Comand;  
}

@Getter  
@Entity  
@Setter  
@NoArgsConstructor  
@AllArgsConstructor  
@Table(name="Start")  
public class Start implements Serializable {  
 @Id  
 @GeneratedValue(strategy = GenerationType.AUTO)  
 @Column(name = "id\_start")  
 private int id\_Start;  
 @Column(name = "name\_start")  
 private String Name\_Start;  
 @Column(name = "about\_start")  
 private String About\_Start;  
 @ManyToOne  
 @JoinColumn(name = "id\_creatst")  
 private User id\_CreatSt;  
  
}

@Entity  
@Getter  
@Setter  
@AllArgsConstructor  
@NoArgsConstructor  
@Table(name="Users")  
public class User implements Serializable {  
 @Id  
 @GeneratedValue(strategy = GenerationType.AUTO)  
 @Column(name = "id\_user")  
 public int id\_user;  
 @Column(name = "user\_name")  
 public String user\_name;  
 @Column(name = "user\_surname")  
 public String user\_surname;  
 @Column(name = "user\_patronic")  
 public String user\_patronic;  
 @Column(name = "user\_login")  
 public String user\_login;  
 @Column(name = "user\_password")  
 public String user\_password;  
 @Column(name = "user\_about")  
 public String user\_about;  
 @Column(name = "user\_messager")  
 public String user\_messeger;  
  
}

@Repository  
  
public interface BundleCSRepository extends JpaRepository<BundleCS,Long> {  
 @Query(value="UPDATE bundlecs SET status\_bundle = :status\_bundle WHERE id\_bundle = :id\_bundle", nativeQuery = true)  
 void Change(@Param("status\_bundle") String status\_bundle, @Param("id\_bundle") long id\_bundle);  
 @Query(value = "SELECT \* FROM bundlecs WHERE id\_start = :id\_start", nativeQuery = true)  
 List<BundleCS> findByIDStart(@Param("id\_start") long id\_start);  
  
}

@Repository  
  
public interface CommandRepository extends JpaRepository<Command,Long> {  
  
 @Query(value="UPDATE command SET name\_command = :name\_Comand, about\_command = :about\_Comand, exp\_command = :exp\_Comand, contact\_command = :contact\_Comand, link\_command = :link\_Comand WHERE id\_command = :id\_Comand", nativeQuery = true)  
 void Change(@Param("name\_Comand") String name\_Comand,@Param("about\_Comand") String About\_Comand,@Param("exp\_Comand") String Exp\_Comand,@Param("contact\_Comand") String Contact\_Comand,@Param("link\_Comand") String Link\_Comand,@Param("id\_Comand") long id\_Comand);  
 @Query(value = "SELECT \* FROM command WHERE id\_creater = :id\_user", nativeQuery = true)  
 List<Command> findByIDUser(@Param("id\_user") long id\_user);  
  
}

@Repository  
public interface StartRepository extends JpaRepository<Start,Long> {  
  
 @Query(value="UPDATE start SET name\_start = :name\_start, about\_start = :about\_start WHERE id\_start = :id\_start", nativeQuery = true)  
 void Change(@Param("name\_start") String name\_start, @Param("about\_start") String about\_start, @Param("id\_start") long id\_start);  
 @Query(value = "SELECT \* FROM start WHERE id\_creatst = :id\_user", nativeQuery = true)  
 List<Start> findByIDStart(@Param("id\_user") long id\_user);  
  
  
}

@Repository  
public interface UserRepository extends JpaRepository<User,Long> {  
  
 @Query(value = "SELECT \* FROM users WHERE users.user\_login = :user\_login AND users.user\_password = :user\_password", nativeQuery = true)  
 List<User> findByNameAndPassword(@Param("user\_login") String user\_login,@Param("user\_password") String user\_password);  
  
 @Modifying  
 @Transactional  
 @Query(value="UPDATE users SET user\_patronic = :user\_patronic, user\_surname = :user\_surname, user\_name = :user\_name, user\_login = :user\_login, user\_about = :user\_about, user\_password = :user\_password, user\_messager = :user\_messager WHERE id\_user = :id\_user", nativeQuery = true)  
 void Change(@Param("user\_patronic") String user\_patronic, @Param("user\_surname") String user\_surname, @Param("user\_name") String user\_name, @Param("user\_login") String user\_login, @Param("id\_user") long id\_user,@Param("user\_password") String user\_password,@Param("user\_about") String user\_about,@Param("user\_messager") String user\_messager);  
  
  
}

@Service  
public class BundleCSService {  
 @Autowired  
 private BundleCSRepository bundleCSRepository;  
 public void save(BundleCS bundleCS){  
 bundleCSRepository.save(bundleCS);  
 }  
 public List<BundleCS> getall(){  
 List<BundleCS> user = bundleCSRepository.findAll();  
 return user;  
 }  
 public void change(String status\_bundle, long id\_bundlecs){  
 bundleCSRepository.Change( status\_bundle, id\_bundlecs);  
 }  
 public List<BundleCS> getIDStart(long id\_start){  
 return bundleCSRepository.findByIDStart(id\_start);  
 }  
 public void delete(long id\_bundle){  
 bundleCSRepository.deleteAllById(Collections.singleton(id\_bundle));  
 }  
}

@Service  
public class BundleUCService {  
 @Autowired  
 public BundleUCRepository bundleUCRepository;  
 public void create(BundleUC bundleUC){  
 bundleUCRepository.save(bundleUC);  
 }  
 public List<BundleUC> getAll(){  
 return bundleUCRepository.findAll();  
 }  
 public void deletebundle(long id\_bundleuc){  
 bundleUCRepository.deleteAllById(Collections.singleton(id\_bundleuc));  
 }  
  
}

@Service  
public class CommandService {  
 @Autowired  
 public CommandRepository commandRepository;  
 public void save(Command command){  
 commandRepository.save(command);  
 }  
 public List<Command> getall(){  
 List<Command> command = commandRepository.findAll();  
 return command;  
 }  
 public Optional<Command> getCommand(long id\_command){  
 Optional<Command> user=commandRepository.findById(id\_command);  
 return user;  
 }  
 public void changeComm(String name\_Comand, String About\_Comand,String Exp\_Comand,String Contact\_Comand,String Link\_Comand,long id\_Comand){  
 commandRepository.Change(name\_Comand, About\_Comand, Exp\_Comand, Contact\_Comand, Link\_Comand, id\_Comand);  
 }  
 public List<Command> getCommID(long id\_user){  
 List<Command> comm = commandRepository.findByIDUser(id\_user);  
 return comm;  
 }  
 public void deleteComm(long id\_command){  
 commandRepository.deleteAllById(Collections.singleton(id\_command));  
 }  
}

@Service  
public class StartService {  
 @Autowired  
 public StartRepository startRepository;  
 public void save(Start start){  
 startRepository.save(start);  
 }  
 public List<Start> getall(){  
 List<Start> start = startRepository.findAll();  
 return start;  
 }  
 public Optional<Start> getStart(long id\_Start){  
 Optional<Start> start=startRepository.findById(id\_Start);  
 return start;  
 }  
 public void deleteStart(long id\_Start){  
 startRepository.deleteAllById(Collections.singleton(id\_Start));  
 }  
 public void changeStart(String name\_start,String about\_start, long id\_start ){  
 startRepository.Change(name\_start,about\_start, id\_start);  
 }  
 public List<Start> findByidStart(long id\_user){  
 return startRepository.findByIDStart(id\_user);  
 }  
}

@Service  
public class UserService {  
  
 @Autowired  
 private UserRepository userRepository;  
  
 public void save(User user){  
 userRepository.save(user);  
 }  
 public List<User> auth(String user\_name, String user\_password){  
 List<User> user = userRepository.findByNameAndPassword(user\_name, user\_password);  
 return user;  
 }  
 public List<User> getall(){  
 List<User> user = userRepository.findAll();  
 return user;  
 }  
 public Optional<User> getUser(long id\_user){  
 Optional<User> user=userRepository.findById(id\_user);  
 return user;  
 }  
 public void deleteUser(long id\_user){  
 userRepository.deleteAllById(Collections.singleton(id\_user));  
 }  
 public void changeUser(String user\_name,String user\_surname, String user\_patronic, long id\_user,String user\_messager, String user\_password,String user\_about, String user\_login ){  
 userRepository.Change(user\_patronic,user\_surname,user\_name,user\_login, id\_user, user\_messager,user\_password,user\_about);  
 }  
}

@SpringBootApplication  
public class BackendApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.run(BackendApplication.class, args);  
 }  
  
}

# **Приложение Б – Листинг программного кода клиентская части**

ReactDOM.createRoot(document.getElementById('root')).render(

  <React.StrictMode>

    <BrowserRouter>

      <Headers />

      <App />

    </BrowserRouter>

  </React.StrictMode>,

)

export default function App() {

  const [auth, setauth] = useState();

  useEffect(()=>{

    let aith = isAuth();

    setauth(aith);

    console.log(aith);

  },[])

  console.log(auth);

  if(auth == true)  return(<RouterAuthYes />)

  else if(auth == false) return (<RouterAuthNo />)

  else return(<RouterAuthNo />)

}

export default function RouterAuthYes(){

    return(

    <Routes>

        <Route path="/" element={<MainPage />} />

        <Route path="/profileuser" element={<ProfileUser />} />

        <Route path="/profilestart/\*" element={<ProfileStarts />} />

        <Route path="/allstarts" element={<AllStarts />} />

        <Route path="/allcmd" element={<AllCmd />} />

        <Route path="/addstarts" element={<AddStart />} />

        <Route path="/settings" element={<SettingsPages />}></Route>

        <Route path="/profilecmd/\*" element={<ProfileCMD />}></Route>

        <Route path="/addcmd" element={<Addcmd />}></Route>

        <Route path="\*" element={<NotFound />} />

    </Routes>

    )

}

export default function RouterAuthNo(){

    return(

    <Routes>

        <Route path="/" element={<MainPage />} />

        <Route path="/auth" element={<Auth />} />

        <Route path="/reg" element={<Register />} />

        <Route path="\*" element={<NotFound />} />

    </Routes>

    )

}

export default function SettingsPages(){

    const [name, setName] = useState();

    const [surname, setSur] = useState();

    const [patronic, setPatr] = useState();

    const [password, setPass] = useState();

    const [Login, setLogin] = useState();

    const [About, setAbout] = useState();

    const [messeger, setMess] = useState();

    const [Vis, setVis]= useState(false);

    const [Mess, setM]= useState();

    const nav = useNavigate();

    const data ={headers:{

        user\_name: name,

        user\_surname: surname,

        user\_patronic: patronic,

        user\_login: Login,

        user\_password: password,

        user\_about: About,

        user\_messager: messeger

    }}

    const data2 = {headers:{

        id\_user: localStorage.getItem("id\_user")

    }}

    const datajs = JSON.stringify(data)

    const onBut = async (e)=>{

        e.preventDefault();

        console.log(datajs)

        if(Login  && name && surname && patronic && password && About && messeger){

            await axios.get("http://localhost:8082/user/change?id\_user="+ parseInt(localStorage.getItem("id\_user")), data)

            .then(function (response) {

                if(response.data === "OK") {

                    setM("Успешно зарегистрировались. Переношу на главную страницу")

                    setVis(true);

                    setTimeout(()=>{nav("/")}, 2500);}

                    else{

                        setM("Неполадки с сервером, Попробуйте позже!")

                        setVis(true);

                        setTimeout(()=>{nav("/")}, 2500);

                    }

            })

        } else {

            setM("Заполните все поля!")

            setVis(true);

        }

    }

    const onDel= ()=>{

        axios.get("http://localhost:8082/user/delete",data2)

        .then((response)=>{console.log(response.data);})

        localStorage.clear();

        localStorage.setItem("isAuth", "false");

        nav("/");

        location.reload();}

    const Reg = ()=>{

        return(

            <div>

                <h3>{Mess}</h3>

            </div>

        )

    }

    return(

        <div className="Setting">

            <h1>Настройки Аккаунта</h1>

            <div>

                <form>

                <button className="btnSet" onClick={(e)=>{onDel(e);}}>Удалить аккаунт</button>

                <h1>Изменение данных</h1>

                <input type="text" placeholder="Введите Фамилию" onChange={(e)=>{setSur(e.target.value)}}/>

                <input type="text" placeholder="Введите Имя" onChange={(e)=>{setName(e.target.value)}}/>

                <input type="text" placeholder="Введите Отчество" onChange={(e)=>{setPatr(e.target.value)}}/>

                <input type="text" placeholder="Введите свой логин" onChange={(e)=>{setLogin(e.target.value)}}/>

                <input type="text" placeholder="Введите пароль" onChange={(e)=>{setPass(e.target.value)}}/>

                <input type="text" placeholder="Расскажите немного о себе " onChange={(e)=>{setAbout(e.target.value)}}/>

                <input type="text" placeholder="Контактные данные" onChange={(e)=>{setMess(e.target.value)}}/>

                <button className="btnSetd" onClick={(e)=>{onBut(e);}}>Изменить данные</button>

                {Vis && <Reg />}

                </form>

            </div>

        </div>

    )

}

export default function Register(){

    const nav = useNavigate();

    const [password, setPass] = useState();

    const [Login, setLogin] = useState();

    const [name, setName] = useState();

    const [surname, setSur] = useState();

    const [patronic, setPatr] = useState();

    const [About, setAbout] = useState();

    const [messeger, setMess] = useState();

    const [Vis, setVis]= useState(false);

    const [Mess, setM]= useState();

    const data ={

        user\_name: name,

        user\_surname: surname,

        user\_patronic: patronic,

        user\_login: Login,

        user\_password: password,

        user\_about: About,

        user\_messeger: messeger

    }

    const onBut = async (e)=>{

        e.preventDefault();

        if(Login && name && surname && patronic && About && messeger && password){

            await axios.post("http://localhost:8082/user/reg", data)

            .then(function (response) {

                if(response.data === "OK") {

                    setM("Успешно зарегистрировались. Переношу на главную страницу")

                    setVis(true);

                    setTimeout(()=>{nav("/")}, 2500);}

                    else{

                        setM("Неполадки с сервером, Попробуйте позже!")

                        setVis(true);

                        setTimeout(()=>{nav("/")}, 2500);

                    }

            })

        } else {

            setM("Заполните все поля!")

            setVis(true);

        }

    }

    const Reg = ()=>{

        return(

            <div>

                <h3>{Mess}</h3>

            </div>

        )

    }

    return(

        <div>

            <form>

                <input type="text" placeholder="Введите Фамилию" onChange={(e)=>{setSur(e.target.value)}}/>

                <input type="text" placeholder="Введите Имя" onChange={(e)=>{setName(e.target.value)}}/>

                <input type="text" placeholder="Введите Отчество" onChange={(e)=>{setPatr(e.target.value)}}/>

                <input type="text" placeholder="Введите свой логин" onChange={(e)=>{setLogin(e.target.value)}}/>

                <input type="text" placeholder="Введите пароль" onChange={(e)=>{setPass(e.target.value)}}/>

                <input type="text" placeholder="Расскажите немного о себе " onChange={(e)=>{setAbout(e.target.value)}}/>

                <input type="text" placeholder="Контактные данные" onChange={(e)=>{setMess(e.target.value)}}/>

                <button id="btnCrAdd"  onClick={(e)=>{onBut(e);}}>Зарегистрироваться</button>

            </form>

            {Vis && <Reg />}

        </div>

    )

}

export default function ProfileUser(){

        const [response, setRes] = useState([]);

        const [Cresponse, setCS] = useState([]);

        const nav = useNavigate();

        const data = {headers: {id\_user: localStorage.getItem("id\_user")}}

        const [arrayStart, setAS] = useState([]);

        useEffect( ()=>{

             axios.get("http://localhost:8082/user/profile",data)

            .then((response)=>{setRes(response.data);});

             axios.get("http://localhost:8082/start/getuserstart", data)

            .then((response)=>{

                console.log(response);setAS(response.data);});

            axios.get("http://localhost:8082/cmd/all")

            .then((response)=>{setCS(response.data);})

        },[])

        const ProfileInfo =()=>{

            return(

                <div className="ProfileInfo">

                    <a>NickName:{response.user\_login}<br /></a>

                    <a>ФИО: {response.user\_surname+" "+response.user\_name +" "+ response.user\_patronic}<br /></a>

                    <a>О себе: {response.user\_about}<br /> </a>

                    <a> Контактные данные: {response.user\_messeger}</a>

                </div>

            )

        }

        console.log(Cresponse)

        const OneStart=(props)=>{

            return(

                <button className="btnAbout" onClick={()=>{nav("/profilestart/"+props.id);}}> <div className="OneStart">

                    <h3> Название: {props.name}</h3>

                    <h4> Руководитель: {props.creater}</h4>

                </div></button>

            )

        }

        const CMD =()=>{

            return(<div className="ArrayStart">{

            Cresponse.map((e) =>{

            if(e.id\_creater?.id\_user ==localStorage.getItem("id\_user") )

            return(

                <div>

                    <button className="btnAbout" onClick={()=>{nav("/profilecmd/"+e.id\_Comand);}}> <div className="OneStart">

                <h3> Название: {e.name\_Comand}</h3>

                <h4> Руководитель: {e.id\_creater?.user\_name}</h4>

            </div></button>

                </div>

            )})}

            </div>)

        }

        const ArrayStart = ()=>{

            return(

                <div className="ArrayStart">

                    {

                        arrayStart.map(function(one){

                            return(

                            <OneStart id={one.id\_Start} name={one.name\_Start} creater={one.id\_CreatSt?.user\_name}/>)

                        })

                    }

                </div>

            )

        }

    return(

        <>

            <h1>Ваш профиль</h1>

            <ProfileInfo />

            <h1>Ваша Команда</h1>

            <CMD />

            <h1>Ваши Старты</h1>

            <ArrayStart />

        </>

    )

}

export default function ProfileStarts(){

    const [ArrInfo,setAI] = useState([])

    const href = parseInt(location.pathname.match(/\d+/));

    const [Vis, setVis]= useState(false);

    const [Mess, setM]= useState();

    const [NameStart, setNS]= useState();

    const [AboutStart, setAS]= useState();

    const [InfoComm, setIC]= useState([]);

    const [ArrBundle, setAB] = useState([]);

    const data = {headers: {

        id\_Start: href

    }}

    const data2 ={headers:{

        name\_Start: NameStart,

        about\_Start: AboutStart}

    }

    const data3 = {headers:{

        id\_Start: ArrInfo.id\_Start

    }}

    const data4 = {

            id\_Start: {

                id\_Start: ArrInfo.id\_Start,

                id\_CreatSt: {

                    id\_user: ArrInfo.id\_CreatSt?.id\_user,

                    user\_name: ArrInfo.id\_CreatSt?.user\_name,

                    user\_surname: ArrInfo.id\_CreatSt?.user\_surname,

                    user\_patronic: ArrInfo.id\_CreatSt?.user\_patronic,

                    user\_login: ArrInfo.id\_CreatSt?.user\_login,

                    user\_password:ArrInfo.id\_CreatSt?.user\_password,

                    user\_about: ArrInfo.id\_CreatSt?.user\_about,

                    user\_messeger: ArrInfo.id\_CreatSt?.user\_messeger

                },

                about\_Start: ArrInfo.about\_Start,

                name\_Start: ArrInfo.name\_Start

            },

            id\_Command: {

                id\_Comand: InfoComm[0]?.id\_Comand,

                name\_Comand: InfoComm[0]?.name\_Comand,

                id\_creater: {

                    id\_user: InfoComm[0]?.id\_creater?.id\_user,

                    user\_name: InfoComm[0]?.id\_creater?.user\_name,

                    user\_surname: InfoComm[0]?.id\_creater?.user\_surname,

                    user\_patronic:InfoComm[0]?.id\_creater?.user\_patronic,

                    user\_login: InfoComm[0]?.id\_creater?.user\_login,

                    user\_password: InfoComm[0]?.id\_creater?.user\_password,

                    user\_about: InfoComm[0]?.id\_creater?.user\_about,

                    user\_messeger: InfoComm[0]?.id\_creater?.user\_messeger

                },

                link\_Comand: InfoComm[0]?.link\_Comand,

                about\_Comand: InfoComm[0]?.about\_Comand,

                exp\_Comand: InfoComm[0]?.exp\_Comand,

                contact\_Comand: InfoComm[0]?.contact\_Comand

            },

            status\_Bundle: "В Ожиданий"

    }

    const data5 ={

        headers:{

            id\_CMD:  localStorage.getItem("id\_user")

        }

    }

    const nav = useNavigate();

    useEffect(()=>{

        axios.get("http://localhost:8082/start/profile",data)

        .then((res)=>{

            setAI(res.data);

        })

        axios.get("http://localhost:8082/cmd/commuser",data5)

        .then((res)=>{

            setIC(res.data);

        })

        axios.get("http://localhost:8082/bundlecs/listbundle",data)

        .then((res)=>{

            console.log(res.data)

            setAB(res.data);

        })

    },[])

    const Reg = ()=>{

        return(

            <div>

                <h3>{Mess}</h3>

            </div>

        )

    }

    const onDel = () =>{

        axios.get("http://localhost:8082/start/delete",data3)

        .then((response)=>{console.log(response.data);})

        nav("/");

    }

    const onCH = ()=>{

        if(NameStart && AboutStart){

            axios.get("http://localhost:8082/start/change?id\_Start="+ArrInfo.id\_Start, data2)

            .then(function (response) {

                if(response.data === "OK") {

                    setM("Успешно добавили.")

                    setVis(true);

                    setTimeout(()=>{location.reload();}, 2500);}

                    else{

                        setM("Неполадки с сервером, Попробуйте позже!")

                        setVis(true);

                        setTimeout(()=>{nav("/")}, 2500);

                    }

            })

        } else {

            setM("Заполните все поля!")

            setVis(true);

        }

        location.reload();

    }

    const onCHbundle = (id,status)=>{

        console.log(id)

        console.log(status)

        axios.get("http://localhost:8082/bundlecs/change?id\_bundlecs="+id, {headers:{ status\_bundle:status}})

        .then(function (response) {

            if(response.data === "OK") {

                location.reload();

            }

        })

        location.reload();

    }

    const onBut = ()=>{

            axios.post("http://localhost:8082/bundlecs/create", data4)

            .then(function (response) {

                if(response.data === "OK") {

                    location.reload();

                }

            })

    }

    const chekbundle=()=>{

        var lol=0

        ArrBundle.map((One)=>{

            if(localStorage.getItem("id\_user") == One.id\_Command.id\_creater?.id\_user)

            {lol=lol+1}

        })

        if (lol==0) return true

        else false

    }

    const onDelBundle=(id)=>{

        axios.get("http://localhost:8082/bundlecs/delete", {headers:{ id\_bundle:id}})

        .then(function (response) {

            if(response.data === "OK") {

                location.reload();

            }

        })

    }

    const TableRes = ()=>{

        return(

            <table>

                <tr>

                    <th>Название команды</th>

                    <th>Ссылка на портфолио</th>

                    <th>Контакты</th>

                    <th>Cтатус</th>

                </tr>

                {

                    ArrBundle.map((One)=>{

                        return(

                        <tr>

                            <td>{One.id\_Command?.name\_Comand}</td>

                            <td>{One.id\_Command?.link\_Comand}</td>

                            <td>{One.id\_Command?.contact\_Comand}</td>

                            <td>{One.status\_Bundle == "approved"? "Одобрено": One.status\_Bundle}</td>

                            <td style={(ArrInfo.id\_CreatSt?.id\_user != localStorage.getItem("id\_user"))? {display:"none"}:{} }>

                            <button id="btnCrAdd" onClick={()=>{nav("/profilecmd/"+One.id\_Command?.id\_Comand); location.reload();}}>Check</button>

                            <button id="btnCrAdd" onClick={()=>{onCHbundle(One.id\_Bundle, "approved")}}>+</button>

                            <button className="btnSe" onClick={()=>{onDelBundle(One.id\_Bundle)}}>-</button></td>

                            <td style={(One.id\_Command.id\_creater.id\_user != localStorage.getItem("id\_user"))? {display:"none"}:{} }><button className="btnSe" onClick={()=>{onDelBundle(One.id\_Bundle)}}>-</button></td>

                        </tr>)

                    })

                }

            </table>

        )

    }

    if(ArrInfo) return(

        <div className="ProfileStarts">

            <h1>Профиль Старта</h1>

            <div className="ProfileInfo">

            <h1>{ArrInfo.name\_Start}</h1>

            <a>Создатель :{ArrInfo.id\_CreatSt?.user\_name}<br /></a>

            <a>Контакт :{ArrInfo.id\_CreatSt?.user\_messeger}<br /></a>

            <a>О cтарте: {ArrInfo.about\_Start} <br /> </a>

        </div>

            <div className="OnlyCreater" style={(ArrInfo.id\_CreatSt?.id\_user != localStorage.getItem("id\_user"))? {display:"none"}:{} }>

                <button className="btnSet" onClick={(e)=>{e.preventDefault();onDel();}}>Удалить</button>

                <form>

                <h1>Изменить данные</h1>

                <input type="text" placeholder="Название старта" onChange={(e)=>{setNS(e.target.value)}}></input>

                <input type="text" placeholder="О старте" onChange={(e)=>{setAS(e.target.value)}}></input>

                {Vis && <Reg />}

                <button  id="btnCrAdd" onClick={(e)=>{e.preventDefault();onCH();}}>Изменить</button>

                </form>

            </div>

            <div className="OnlyCMD" style={(ArrInfo.id\_CreatSt?.id\_user != localStorage.getItem("id\_user")) &&InfoComm[0] && chekbundle()?{}:{display:"none"} }>

                <button  id="btnCrAdd" onClick={(e)=>{e.preventDefault();onBut();}}>Откликнуться</button>

            </div>

            <TableRes />

        </div>

    )

    else {nav("/")}

}

export default function ProfileCMD(){

    const [ArrInfo,setAI] = useState([]);

    const [PartInfo,setPI] = useState([]);

    const [Name,setName] = useState();

    const [Surname,setSurName] = useState();

    const [Role,setRole] = useState();

    const href = parseInt(location.pathname.match(/\d+/));

    const [Vis, setVis]= useState(false);

    const [Vis2, setVis2]= useState(false);

    const [Mess, setM]= useState();

    const [Mess2, setM2]= useState();

    const [NameComand,setNC] = useState();

    const [AboutCommand,setAC] = useState();

    const [ExpComand,setEC] = useState();

    const [ContactComand,setCC] = useState();

    const [PortComand,setPC] = useState();

    const nav = useNavigate();

    const data = {headers: {

        id\_CMD: href

    }}

    const data2 =

    {

            name\_Part: Name,

            surname\_Part: Surname,

            role\_Part: Role,

            id\_CommandCR: {

                id\_Comand: ArrInfo.id\_Comand,

                name\_Comand:  ArrInfo.name\_Comand,

                id\_creater: {

                    id\_user:ArrInfo.id\_creater?.id\_user ,

                    user\_name: ArrInfo.id\_creater?.user\_name,

                    user\_surname: ArrInfo.id\_creater?.user\_surname,

                    user\_patronic: ArrInfo.id\_creater?.user\_patronic ,

                    user\_login: ArrInfo.id\_creater?.user\_login,

                    user\_password: ArrInfo.id\_creater?.user\_password,

                    user\_about: ArrInfo.id\_creater?.user\_about,

                    user\_messeger: ArrInfo.id\_creater?.user\_messeger

                },

                about\_Comand:  ArrInfo.about\_Comand,

                link\_Comand: ArrInfo.link\_Comand,

                exp\_Comand: ArrInfo.exp\_Comand ,

                contact\_Comand: ArrInfo.contact\_Comand

            }

    }

    const data3 = {headers:{

        id\_command: ArrInfo.id\_Comand

    }}

    const data4 ={

        headers :{

            name\_Comand: NameComand,

            about\_Comand: AboutCommand,

            exp\_Comand:ExpComand,

            contact\_Comand:ContactComand,

            link\_Comand: PortComand

        }

    }

    useEffect(()=>{

        axios.get("http://localhost:8082/cmd/prof",data)

        .then((res)=>{

            console.log(res);

            setAI(res.data);

        })

        axios.get("http://localhost:8082/bundleuc/all",data)

        .then((res)=>{

            console.log(res);

            setPI(res.data);

        })

    },[])

    const  onBut=()=>{

        if(Name && Surname && Role){

            axios.post("http://localhost:8082/bundleuc/create", data2)

            .then(function (response) {

                if(response.data === "OK") {

                    setM("Успешно добавили.")

                    setVis(true);

                    setTimeout(()=>{location.reload();}, 2500);}

                    else{

                        setM("Неполадки с сервером, Попробуйте позже!")

                        setVis(true);

                        setTimeout(()=>{nav("/")}, 2500);

                    }

            })

        } else {

            setM("Заполните все поля!")

            setVis(true);

        }

    }

    const delPart=(id)=>{

        axios.get("http://localhost:8082/bundleuc/delete?id\_bundleuc="+id)

        location.reload();

    }

    const InDel = (props)=>{

        if(ArrInfo.id\_creater?.id\_user == localStorage.getItem("id\_user"))

        return(

            <button className="btnSet" onClick={(e)=>{e.preventDefault();delPart(props.id)}}>Удалить</button>

    )

    }

    const TablePers = ()=>{

        return(

            <table>

                <tr>

                    <th>

                        Фамилия

                    </th>

                    <th>

                        Имя

                    </th>

                    <th>

                       Роль

                    </th>

                </tr>

                    {

                        PartInfo.map((Part)=>{

                            if(ArrInfo.id\_Comand === Part.id\_CommandCR?.id\_Comand){

                            return(

                                <tr>

                                    <td>{Part.name\_Part}</td>

                                    <td>{Part.surname\_Part}</td>

                                    <td>{Part.role\_Part}</td>

                                     <td><InDel id={Part.id\_Bundle}/></td>

                                </tr>

                            )}

                    })

                    }

            </table>

        )

    }

    const onDel= ()=>{

        axios.get("http://localhost:8082/cmd/delete",data3)

        .then((response)=>{console.log(response.data);})

        nav("/");}

    const ButDe= ()=>{

        if(ArrInfo.id\_creater?.id\_user == localStorage.getItem("id\_user"))

        return(

            <button id="ButDe" className="btnSet" onClick={()=>{onDel()}}>Удалить команду</button>

    )

    }

    const Reg = ()=>{

        return(

            <div>

                <h3>{Mess}</h3>

            </div>

        )

    }

    const Reg2= ()=>{

        return(

            <div>

                <h3>{Mess2}</h3>

            </div>

        )

    }

    const onCH = ()=>{

        if(NameComand && AboutCommand && ExpComand && ContactComand&&PortComand){

            axios.get("http://localhost:8082/cmd/change?id\_Comand="+ArrInfo.id\_Comand, data4)

            .then(function (response) {

                if(response.data === "OK") {

                    setM2("Успешно изменили.")

                    setVis2(true);

                    setTimeout(()=>{location.reload();}, 1500);}

                    else{

                        setM2("Неполадки с сервером, Попробуйте позже!")

                        setVis2(true);

                        setTimeout(()=>{nav("/")}, 1500);

                    }

            })

        } else {

            setM2("Заполните все поля!")

            setVis2(true);

        }

        location.reload();

    }

    return(<div className="ProfileCMD">

        <h1> Профиль Команды</h1>

        <div className="ProfileInfo">

            <a>Название команды:{ArrInfo.name\_Comand}<br /></a>

            <a>Создатель:{ArrInfo.id\_creater?.user\_name}<br /></a>

            <a>О команде: {ArrInfo.about\_Comand} <br /> </a>

            <a> Контактные данные: {ArrInfo.contact\_Comand} <br /></a>

            <a> Наш опыт: {ArrInfo.exp\_Comand} <br /></a>

            <a> Портфолио: {ArrInfo.link\_Comand}</a>

        </div>

        <ButDe />

        <form style={(ArrInfo.id\_creater?.id\_user != localStorage.getItem("id\_user"))? {display:"none"}:{} }>

            <h1>Добавить участника</h1>

                <input type="text" placeholder="Фамилия" value={Surname} onChange={(e)=>{setSurName(e.target.value)}}></input>

                <input type="text" placeholder="Имя" value={Name}  onChange={(e)=>{setName(e.target.value)}}></input>

                <input  type="text" placeholder="Роль" value={Role}  onChange={(e)=>{setRole(e.target.value)}} ></input>

                {Vis && <Reg />}

                <button  id="btnCrAdd" onClick={(e)=>{e.preventDefault(); onBut();}}>Добавить</button>

            <h1>Изменить данные</h1>

                <input type="text" placeholder="Название команды"  onChange={(e)=>{setNC(e.target.value)}}></input>

                <input type="text" placeholder="О команде"   onChange={(e)=>{setAC(e.target.value)}}></input>

                <input  type="text" placeholder="Контактные данные"   onChange={(e)=>{setCC(e.target.value)}} ></input>

                <input type="text" placeholder="Опыт"  onChange={(e)=>{setEC(e.target.value)}}></input>

                <input type="text" placeholder="Ссылка на Портфолио"   onChange={(e)=>{setPC(e.target.value)}}></input>

                {Vis2 && <Reg2 />}

                <button  id="btnCrAdd" onClick={(e)=>{e.preventDefault(); onCH();}}>Изменить</button>

            </form>

        <h1>Участники команды</h1>

        <TablePers />

    </div>);

}

export default function NotFound(){

    const nav = useNavigate();

    return(

        <div className="NotFound">

            <h1 className="img404">Error 404</h1>

            <h1>Страница не найдена!<br /> Вернитесь на главную страницу.</h1>

            <button className="btnNot" onClick={()=>{nav("/");}}>Главная страница</button>

        </div>

    )

}

export default function MainPage(){

    const nav = useNavigate();

    const auth = isAuth();

    const [isVis, setVis] = useState(false);

    console.log(auth);

    const btnCklick=()=>{

    if(auth) return(nav("/allstarts"))

    else setVis(true)

}

    return (

        <div>

        <div className="Main">

           <div className="LogoMain">

                <h1>Prosto Start</h1>

                <p>Легко и просто ищите людей для своего старта</p>

                <button onClick={()=>{btnCklick()}}>Искать</button>

            </div>

            <img src={img} />

        </div>

        {isVis && <Auth />}

        </div>

    )

}

export default function Headers(){

    const nav = useNavigate();

    const [nickUser, setNick]= useState('');

    const [classMenu, setMenu] = useState("container");

    const [visMenu, setVisMenu]= useState(false);

    useEffect(()=>{

        setNick(localStorage.getItem("Nick"))

    },[])

    const onBut = ()=>{

        localStorage.clear();

        localStorage.setItem("isAuth", "false");

        nav("/");

        location.reload();

    }

    const LogoHead = ()=>{

        return(

            <p className="LogoHead">

            <a className="logo">PS.c</a>om

        </p>

        )

    }

    const BtnLogo = ()=>{

        return(

            <div id="idBtn">

            <div className="btnLogo">

            <a>{nickUser}</a>

            <button onClick={()=>{onBut()}}>Выйти</button>

        </div></div>

        )

    }

    const LogoAvatar =()=>{

        if(isAuth()== true)return(

            <div className="MLogo">

            <MenuWeb />

            <BtnLogo />   </div>

        );

        else return( <></>

        );

    }

    const MenuBottom = ()=>{

        return(

            <div id="myDropdown" className="dropdown-content">

            <div className="ColumnMenu"> <a className="listchild" href="/profileuser"> Мой профили </a>

            <a className="listchild" href="/allstarts">Все старты</a>

            <a className="listchild" href="/allcmd">Все команды</a>

            <a className="listchild" href="/settings">Настройки профиля</a>

            </div>

        </div>

        )

    }

    const MenuWeb = ()=>{

        function myFunction() {

            if(classMenu === "container" )

            {

                setMenu("change");

                setVisMenu(true);

            }

            else

            {

                setMenu("container");

                setVisMenu(false);

            }

        }

        return(

            <div id="Menu"  className={classMenu} onClick={()=>{myFunction()}}>

                 <div className="bar1"></div>

                <div className="bar2"></div>

                <div className="bar3"></div>

                {visMenu && <MenuBottom /> }

            </div>

        )

    }

    return(

    <div className="Headers">

        <LogoHead />

        <div>

            <img />

            <LogoAvatar />

        </div>

    </div>)

}

export default function Auth(){

    const [login,setLogin] = useState()

    const [pass,setPass] = useState()

    const [vis,setVis] = useState(false);

    const [Messread, setMR] = useState();

    const data ={headers:{

        user\_login: login,

        user\_password: pass}

    }

    const nav = useNavigate();

    const clickAuth = (e)=>{

        e.preventDefault();

        if(login&&pass){

        axios.get("http://localhost:8082/user/auth",data)

        .then((response)=>{

            console.log( response.data);

            if(response.data[0]){

                localStorage.setItem("Nick", response.data[0].user\_name);

                localStorage.setItem("id\_user", response.data[0].id\_user);

                localStorage.setItem("isAuth", "true");

                setMR("Успешно вошли. Переадресация.")

                setVis(true);

                setTimeout(()=>{nav("/"); location.reload();}, 2500);

            } else {

                setMR("Неправильный логин или пароль")

                setVis(true);

            }

        })

        } else{

            setMR("Заполните все поля.")

            setVis(true);

        }

    }

    const Mess = ()=>{

        return(

            <h4>{Messread}</h4>

        )

    }

    return(

        <div className="AuthPage">

            <img src={img} />

            <div className="FormAuth">

                <h1>Авторизация</h1>

                <form>

                    <div className="inputLogin" >

                        <input type="text" placeholder="Введите логин:" onChange={(e)=>{setLogin(e.target.value)}} />

                    </div>

                    <div  className="inputss">

                        <input type="text" placeholder="Введите пароль:" onChange={(e)=>{setPass(e.target.value)}} />

                    </div>

                    {vis && <Mess />}

                    <p>Нет аккаунта?<button className="btnReg" onClick={()=>{nav("/reg");}}> Создай его!</button></p>

                    <button className="btnAuth" onClick={(e)=>{clickAuth(e)}}>Войти</button>

                </form>

            </div>

        </div>

    )

}

export default function AllStarts(){

    const nav = useNavigate();

    const [arrayStart, setAS] = useState([]);

    useEffect(()=>{

        axios.get("http://localhost:8082/start/all")

        .then((response)=>{setAS(response.data);})

       },[]);

    const OneStart=(props)=>{

        return(

            <button className="btnAbout" onClick={()=>{nav("/profilestart/"+props.id);}}> <div className="OneStart">

                <h3> Название: {props.name}</h3>

                <h4> Руководитель: {props.creater}</h4>

            </div></button>

        )

    }

    const ArrayStart = ()=>{

        return(

            <div className="ArrayStart">

                {

                    arrayStart.map(function(one){

                        console.log(one)

                        return(

                        <OneStart id={one.id\_Start} name={one.name\_Start} creater={one.id\_CreatSt?.user\_name}/>)

                    })

                }

            </div>

        )

    }

    return(

        <div>

            <h1 className="AllLogo">Все Старты</h1>

            <button className="btnAbout" onClick={()=>{nav("/addstarts");}}>+</button>

            <ArrayStart />

        </div>

    )

}

export default function AllCmd(){

    const nav = useNavigate();

    const [arrayStart, setAS] = useState([]);

    const [Vis,setVis] = useState(0);

    useEffect(()=>{

        axios.get("http://localhost:8082/cmd/all")

        .then((response)=>{setAS(response.data);})

       },[]);

       const GoAddcmd =()=>{

        arrayStart.map((One)=>{

        if(One.id\_creater.id\_user==localStorage.getItem("id\_user"))

        setVis(Vis+1);

       })

       if(Vis==0) return(

        <button className="btnAbout" onClick={()=>{nav("/addcmd");}}>+</button>

       )

       }

    return(

        <div className="Allcmd">

            <h1>Все команды</h1>

            <GoAddcmd />

            <table>

                <tr>

                    <th>Название команды</th>

                    <th>Контакт</th>

                    <th>Портфолио</th>

                    <th></th>

                </tr>

                {

                 arrayStart.map((one)=>{

                    return(

                        <tr>

                            <td>{one.name\_Comand}</td>

                            <td>{one.contact\_Comand}</td>

                            <td>{one.link\_Comand}</td>

                            <button className="btnAbout" onClick={()=>{nav("/profilecmd/"+one.id\_Comand)}}>Посмотреть</button>

                        </tr>

                    )

                 })

                }

            </table>

        </div>

    )

}

export default function AddStart(){

 const [nameStart,setNS] = useState("");

 const [AboutStart,setAS] = useState("");

 const [response, setRes] = useState([]);

 const data2 = {headers: {id\_user: localStorage.getItem("id\_user")}}

 const nav = useNavigate();

 const data = {

    name\_Start: nameStart,

    about\_Start: AboutStart,

    id\_CreatSt: {

        id\_user: response.id\_user,

        user\_name: response.user\_name,

        user\_surname: response.user\_surname,

        user\_patronic: response.user\_patronic,

        user\_login: response.user\_login,

        user\_password: response.user\_password,

        user\_about: response.user\_about,

        user\_messeger: response.user\_messeger

    },

 }

 useEffect( ()=>{

    axios.get("http://localhost:8082/user/profile",data2)

   .then((response)=>{setRes(response.data);});},[])

    const onPost = (e)=>{

        e.preventDefault();

        console.log(data);

        axios.post("http://localhost:8082/start/create", data)

        .then((res)=>{console.log(res.data)});

        nav("/allstarts");

    }

    return(

        <div>

            <h1>Добавление Старта</h1>

            <form>

            <div className="inputLogin" >

                <input type="text" placeholder="Введите название стартапа" onChange={(e)=>{setNS(e.target.value)}}></input>

            </div>

            <div className="inputLogin" >

                <input type="text" placeholder="Расскажите о сути стартапа"onChange={(e)=>{setAS(e.target.value)}}></input>

            </div>

                <button id="btnCrAdd" onClick={(e)=>{onPost(e)}}>Создать</button>

            </form>

        </div>

    )

}

export default function Addcmd(){

    const nav = useNavigate();

    const [User, setUser]= useState([]);

    const [NameCMD,  setName] = useState();

    const [aboutCMD, setAbout] = useState();

    const [contactCMD, setContact] = useState();

    const [expCMD, setEXP] = useState();

    const [linkCMD, setLink] = useState();

    const [Vis, setVis]= useState(false);

    const [Mess, setM]= useState();

    const data2 = {headers: {id\_user: localStorage.getItem("id\_user")}}

    const data={

        name\_Comand: NameCMD,

        about\_Comand: aboutCMD,

        exp\_Comand: expCMD,

        contact\_Comand: contactCMD,

        id\_creater: {

            id\_user: User.id\_user,

            user\_name: User.user\_name,

            user\_surname: User.user\_surname,

            user\_patronic: User.user\_patronic,

            user\_login: User.user\_login,

            user\_password: User.user\_password,

            user\_about: User.user\_about,

            user\_messeger: User.user\_messeger

            },

        link\_Comand: linkCMD

    }

    const onBut = async (e)=>{

        axios.get("http://localhost:8082/user/profile",data2)

        .then((response)=>{setUser(response.data);});

        e.preventDefault();

        if(NameCMD && aboutCMD && contactCMD && expCMD && linkCMD){

            await axios.post("http://localhost:8082/cmd/create", data)

            .then(function (response) {

                if(response.data === "OK") {

                    setM("Успешно зарегистрировались. Переношу на главную страницу")

                    setVis(true);

                    setTimeout(()=>{nav("/")}, 2500);}

                    else{

                        setM("Неполадки с сервером, Попробуйте позже!")

                        setVis(true);

                        setTimeout(()=>{nav("/allcmd")}, 2500);

                    }

            })

        } else {

            setM("Заполните все поля!")

            setVis(true);

        }

    }

    const Reg = ()=>{

        return(

            <div>

                <h3>{Mess}</h3>

            </div>

        )

    }

    return(

        <div>

            <h1>Добавление своей команды</h1>

            <form>

                <input type="text" placeholder="Введите Название" onChange={(e)=>{setName(e.target.value)}}/>

                <input type="text" placeholder="Коротко опишите свою команду" onChange={(e)=>{setAbout(e.target.value)}}/>

                <input type="text" placeholder="Контактные данные"  onChange={(e)=>{setContact(e.target.value)}}/>

                <input type="text" placeholder="Расскажите о опыте команды"  onChange={(e)=>{setEXP(e.target.value)}}/>

                <input type="text" placeholder="Ссылка на портфолио команды"  onChange={(e)=>{setLink(e.target.value)}}/>

                <button id="btnCrAdd" onClick={(e)=>{onBut(e);}}>Создать</button>

                {Vis && <Reg />}

            </form>

        </div>

    )

}

export default function isAuth(){

   let auth = localStorage.getItem("isAuth");

      if(auth === "true") return true

      else if(auth === "false") return false

      else localStorage.setItem("isAuth", false);

}