|  |  |  |
| --- | --- | --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** | | |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ ИМЕНИ К.Г. РАЗУМОВСКОГО (ПЕРВЫЙ КАЗАЧИЙ УНИВЕРСИТЕТ)»**  **(ФГБОУ ВО «МГУТУ ИМ. К.Г. РАЗУМОВСКОГО (ПКУ)»)** | | |
|  | | |
| **УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ** | | |
|  | | |
| **ОТЧЁТ** | | |
| ПО УЧЕБНОЙ ПРАКТИКЕ  УП.02.01. Учебная практика  профессионального модуля | | |
| ПМ.02. Осуществление интеграции программных модулей | | |
|  | | |
|  | | |
| студента группы 090207-9о-20/2, обучающегося специальности  09.02.07 «Информационные системы и программирование» | | |
| Асылбек уулу Бакыта | | |
|  | | |
|  | | |
| Период практики | | |
| С «15» июня 2023 г. по «5» июля 2023 г. | | |
|  | | |
|  | | |
| Руководитель практики: Миркитанов Павел Викторович | | |
| Выполнил студент: Асылбек уулу Бакыт | | |
|  | | |
|  | Студент |  |
|  | Руководитель практики |  |
|  | Оценка |  |

Москва

2023

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc139404605)

[ОСНОВНАЯ ЧАСТЬ 5](#_Toc139404606)

[1 Разработка программы 5](#_Toc139404607)

[1.1 Спецификации разрабатываемого продукта 5](#_Toc139404608)

[1.2 Описание проектирования базы данных 7](#_Toc139404609)

[1.3 Описание стадий программного продукта 8](#_Toc139404610)

[2 Практические работы по системе контроля версий Git 11](#_Toc139404611)

[ЗАКЛЮЧЕНИЕ 12](#_Toc139404612)

[ПРИЛОЖЕНИЕ A 13](#_Toc139404613)

[ПРИЛОЖЕНИЕ B 15](#_Toc139404614)

# ВВЕДЕНИЕ

Выполнение заданий УП.02.01. Учебной практики ПМ.02. Осуществление интеграции программных модулей подразумевает освоение ряда умений и компетенций.

Данная учебная практика формирует следующие профессиональные компетенции:

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.

ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

В ходе работы надо получить практический опыт:

* в интеграции модулей в программное обеспечение;
* в отладке программных модулей.

Во время учебной практики должны быть приобретены навыки:

* работы с системой контроля версий Git;
* использование методов для получения кода с заданной функциональностью и степенью качества.

Помимо этого, были получены общие компетенции:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения.

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности

ОК 09. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках.

ОК 11. Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере.

# ОСНОВНАЯ ЧАСТЬ

Данная практика состоит из разработки программного продукта, который будет описан далее, и практических работ по системе контроля версий Git.

## Разработка программы

### Спецификации разрабатываемого продукта

Для хранения личных данных и фотографии пользователя будет использована база данных MSSQL. В ней же будут созданы следующие таблицы:

* сущность «Информация о пользователях» будет включать в себя следующие атрибуты:

1. имя пользователя;
2. фамилия;
3. имя;
4. отчество;
5. день рождения;
6. пароль;
7. пол;
8. почта;
9. вопрос;
10. ответ;

* сущность «Для аватара» будет включать атрибуты, указанные ниже:

1. имя пользователя;
2. ссылка на фото профиля.

Программа будет состоять из следующих окон:

* авторизация пользователей;
* восстановления пароля;
* регистрация новых пользователей;
* окно профиля;
* настройки программного продукта;
* окно редактирования данных;
* окно авиабилеты;
* окно избранное.

Система будет обеспечивать аутентификацию и авторизацию пользователей.

При регистрации личные данные будут вноситься в базу данных, а также по умолчанию пользователь будет иметь классическое фото профиля.

В окне восстановления пароля необходимо будет заполнить поля, чтобы увидеть свой пароль.

В окне профиля будут отображены фото профиля, фамилия, имя и отчество пользователя, а также посты с возможностью увеличения масштаба при нажатии кнопки «Подробнее». К тому же данное окно будет иметь возможность изменения фото профиля и кнопки, при нажатии на которых можно перейти в окна настройки, авиабилеты, избранное, а также присутствует возможность смены пользователя и изменения, сохранение статуса.

В настройках будет находится следующее:

* личная информация пользователя;
* возможность редактирование личной информации;
* ссылка на фото профиля;
* фото профиля;
* смена языка отображения (локализации);
* возможность поменять тему;
* удаление аккаунта;
* возврат на окно профиля.

В настройках можно будет менять язык, кроме окон: профиля, авиабилетов и избранных.

Окно «Авиабилеты» будут содержать в себе фильтр по данным авиабилетам, данные которых будут взяты из JSON-файла. К тому же, при нажатии на ячейку, данные передаются в окно «Избранное» и добавляется возможность приобрести выбранный билет.

В окне «Избранное» можно будет работать только с выбранными Пользователем авиабилетами.

### Описание проектирования базы данных

Физическая модель базы данных состоит из 2 таблиц:

Таблица 1 – «Информация о пользователях» (Рисунок 1)

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибуты | Типы данных | Ограничения | Особенности |
| Имя пользователя | Varchar | 50 символов | Первичный, внешний ключ |
| Фамилия | Varchar | 50 символов | Разрешено быть пустым |
| Имя | Varchar | 50 символов | Разрешено быть пустым |
| Отчество | Varchar | 50 символов | Разрешено быть пустым |
| Дата рождения | Date | Ограничений нет | Разрешено быть пустым |
| Пароль | Char | 8 символов | Разрешено быть пустым |
| Пол | Char | 1 символ | Разрешено быть пустым |
| Почта | Varchar | 50 символов | Разрешено быть пустым |
| Вопрос | Varchar | 50 символов | Разрешено быть пустым |
| Ответ | Varchar | 50 символов | Разрешено быть пустым |

Атрибут «День рождения» имеет указанный тип данных, так как содержит в себе формат информации дата. Ограничения не указаны, так как их нельзя задать в базе данных.

Атрибут «Пароль» имеет ограничение 8 символов, так как он достаточно длинный, чтобы обеспечить достаточную безопасность, и по-прежнему достаточно короткий, чтобы пользователи легко запоминали. А тип данных соответствующий из-за соображения безопасности.

Атрибут «Пол» имеет ограничение в один символ и данный тип данных, потому что так легче и удобно. К тому же этого более чем достаточно для этой информации.

Остальные имеют указанный тип данных, так как в данном учебном заведении учили использовать выбранный тип данных, если формат информации текстовый. При выборе данного типа данных по умолчанию обозначено ограничение в пятьдесят символов для имени пользователя, фамилии, имени, отчества, почты, вопроса и ответа.

Таблица 2 – «Для аватара» (Рисунок 1)

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибуты | Типы данных | Ограничения | Особенности |
| Имя пользователя | Varchar | 50 символов | Первичный, внешний ключ |
| url-адрес фотографии | Varchar | 100 символов | Разрешено быть пустым |

Атрибуты имеют указанный тип данных, так как в данном учебном заведении учили использовать выбранный тип данных, если формат информации текстовый.

При выборе данного типа данных по умолчанию обозначено ограничение в пятьдесят символов для имени пользователя.

Url-адрес имеет длину в два раза больше, так как ссылки могут быть длинными. К тому же без ограничений программа не могла подключиться к базе данных, поэтому и выбрано указанное число.

Таблица 1 имеет связь один к одному с таблицей 2, так как каждый пользователь может отображать только одно фото в своем профиле.

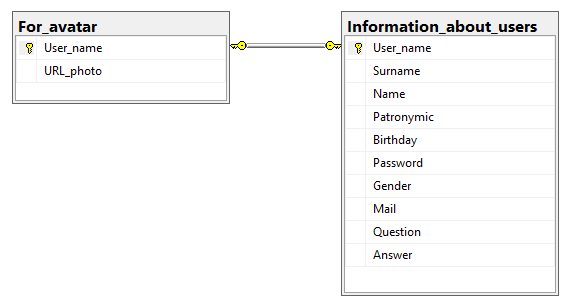


Рисунок 1 - Диаграмма физической модели базы данных

### Описание стадий программного продукта

#### Подключение к базе данных

Для подключения базы данных использована библиотека Entity Framework 6.x, так как оно уменьшает несогласованность между реляционным и объектно-ориентированным мирами, позволяя разработчикам создавать приложения, которые взаимодействуют с данными, хранящимися в реляционных базах данных и устраняя необходимость писать большой объем инфраструктурного кода для доступа к данным.

Этапы подключения базы данных:

1. Нажимаем на «Создать элемент»
2. Выбираем элемент «Модель ADO.NET EDM» и даем ему название.
3. Выбираем «Конструктор EF из базы данных»
4. Создаем подключение, вводим имя сервера и выбираем базу данных.
5. После того как выбрали базу данных, нажимаем «Далее».
6. Выбираем версию Entity Framework, которую будем использовать.
7. Выбираем все таблицы, с которыми будем взаимодействовать через программу, и нажимаем готово.

Для обращения к базе данных используем паттерн SINGLETON.

#### Аутентификация, регистрация и восстановление пароля

При запуске приложения открывается окно с кнопкой «Запустить приложение». После нажатия новый пользователь переходит на окно авторизации (Рисунок 2) и там же можно перейти в окно регистрации (Рисунок 3) или в окно восстановления пароля (Рисунок 4), если пользователь забыл пароль.

Аутентификация (Рисунок 2) пользователей реализована с использованием LINQ выражений, сравнивая заполненные поля с информацией из базы данных.

В регистрации (Рисунок 3) сначала следует нажать на кнопку сохранить, чтобы прочитать все ограничения по вводу данных. Если введенный новым пользователем логин уже существует в базе данных, то выведется сообщение об этом.

В окне восстановления пароля (Рисунок 4) необходимо после того, как заполнили поля, нажимать на кнопку «Проверить» чтобы сверить данные и вывести контрольный вопрос, при правильном ответе на который отображается пароль пользователя.

В случае если пользователь уже зарегистрировался, при нажатии кнопки в главном окне он сразу переноситься на окно профиля (Рисунок 5).

#### Окна профиля, настроек, авиабилетов и избранных

В данном окне отображается фото профиля, ФИО пользователя и посты. Информация, отображаемая в постах, хранится в классе. Также можно увеличить масштаб поста. Для этого следует нажать на кнопку «Подробнее» и окно профиля преобразуется для детализации постов (Рисунок 6). К тому же в этом окне присутствует возможность изменить фото профиля, изменить и сохранить статус, перейти в окна: авиабилеты (Рисунок 7), избранное (Рисунок 8), настройки (Рисунок 9) и авторизация (Рисунок 2).

В окне настройки (Рисунок 9) можно менять язык с русского на английский и наоборот, также с темой светлую на темную. В случае удаления аккаунта его личные данные стираются из базы данных, и пользователя отправляют в окно авторизации. При нажатии на кнопку изменить нас переносит на окно редактирования данных, где можно изменить любую информацию, кроме логина пользователя, поле которого приобретает свойство только для чтения. Следовательно, ее невозможно изменить.

В окне авиабилеты (Рисунок 7) отображается информация о билетах в табличном виде. Таблица может меняться благодаря фильтру. При нажатии на пустую ячейку пользователя переносит на окно «Избранное» (Рисунок 9). После перехода можно появляется возможность приобрести билеты.

Также создан репозиторий на GitHub, куда выложен разработанный проект и проведенное тестирование всего функционала приложения через git, ссылка на которую имеется в приложении.

## Практические работы по системе контроля версий Git

Работ по системе контроля версий всего 3. В каждой практике содержатся по 3 задания.

Практическая работа №1 включает в себя такие задания, как работа с базовыми командами git (Рисунок 10) через приложение git bash, импорт существующего проекта (Рисунок 11) из локального репозитория в удаленную и отмена изменений (Рисунок 12) в случае возникновения ошибок при добавлении файлов с ошибками.

Практическая работа состояла из следующего:

1. Создание веток (Рисунок 13).
2. Слияние изменений (Рисунок 14).
3. Работа с историей изменений и переключение между коммитами (Рисунок 15.1, 15.2).

А в третьей работе необходимо было разрешать конфликты (Рисунок 16.1, 16.2) при выгрузке данных из локального репозитория в удаленный, поработать с системой учета ошибок «issues» и с функцией GitHub «pull-request» позволяющая попросить владельца репозитория, от которого мы сделали fork, загрузить наши изменения обратно в свой репозиторий.

# ЗАКЛЮЧЕНИЕ

В результате прохождения учебной практики был получен практический опыт в интеграции модулей в программное обеспечение, в тестировании и отладке программных модулей, а также были приобретены навыки работы с системой контроля версий Git, использование методов для получения кода с заданной функциональностью и степенью качества.

Помимо этого, был получен опыт работы с коллективом, способы выполнения профессиональных задач, оценены их эффективность и качество. Приобретен навык осуществление поиска и использования информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития, использование информационно-коммуникационных технологий в профессиональной деятельности.

Учебная практика позволила закрепить и углубить знания в сфере осуществления интеграции программных модулей.

# ПРИЛОЖЕНИЕ A

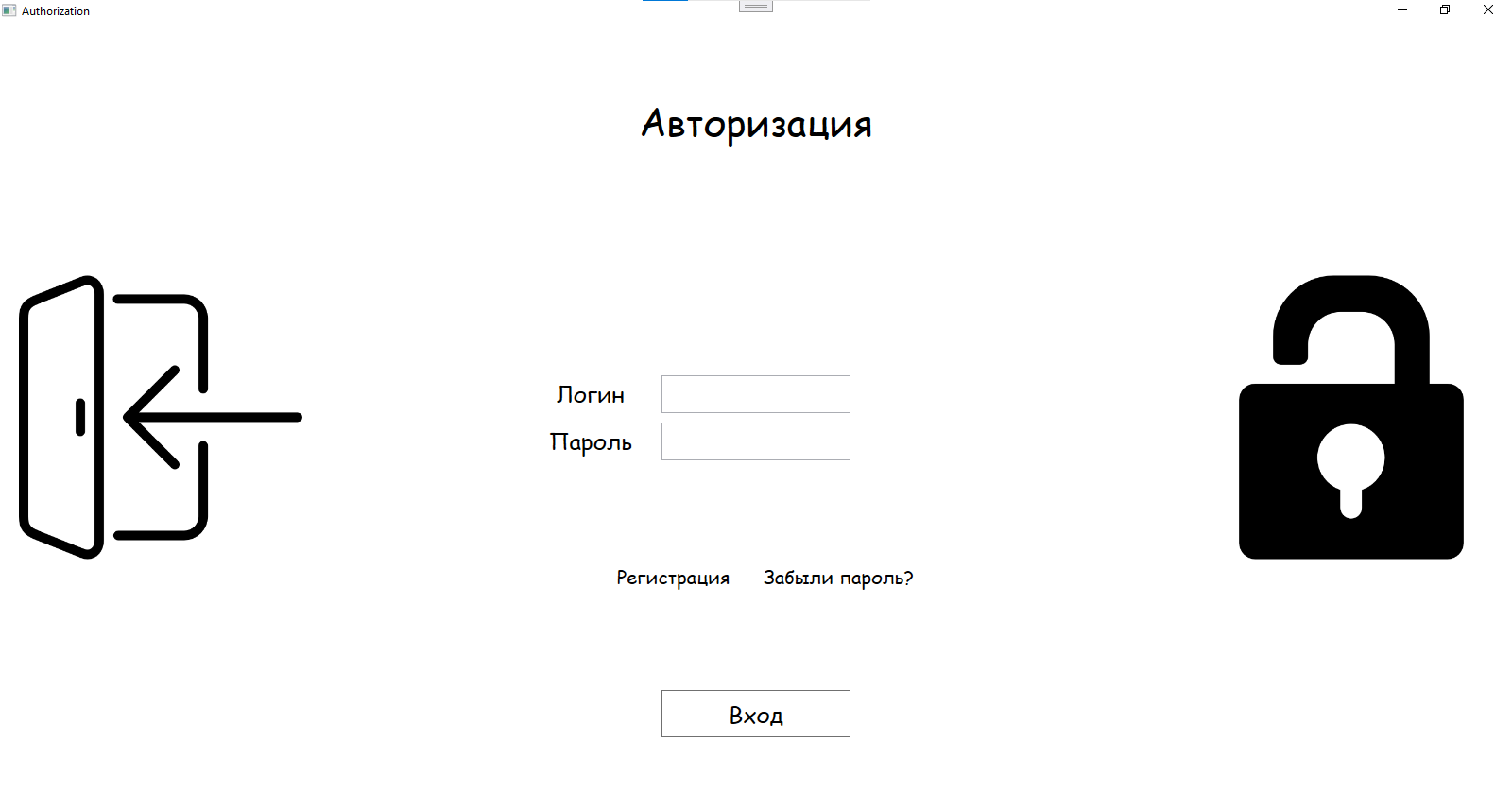


Рисунок 2 - Окно «Авторизация»

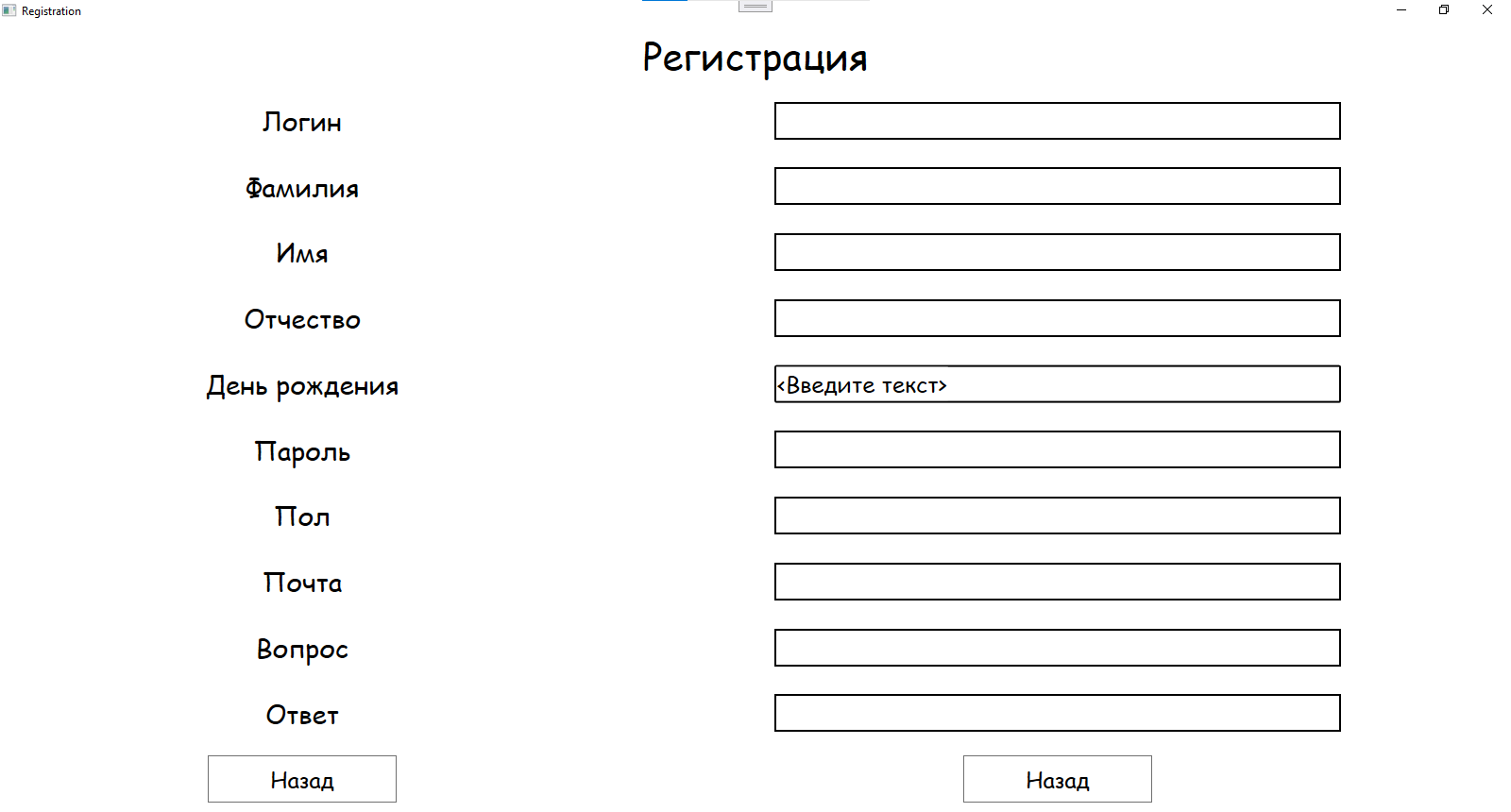


Рисунок 3 - Окно «Регистрация»

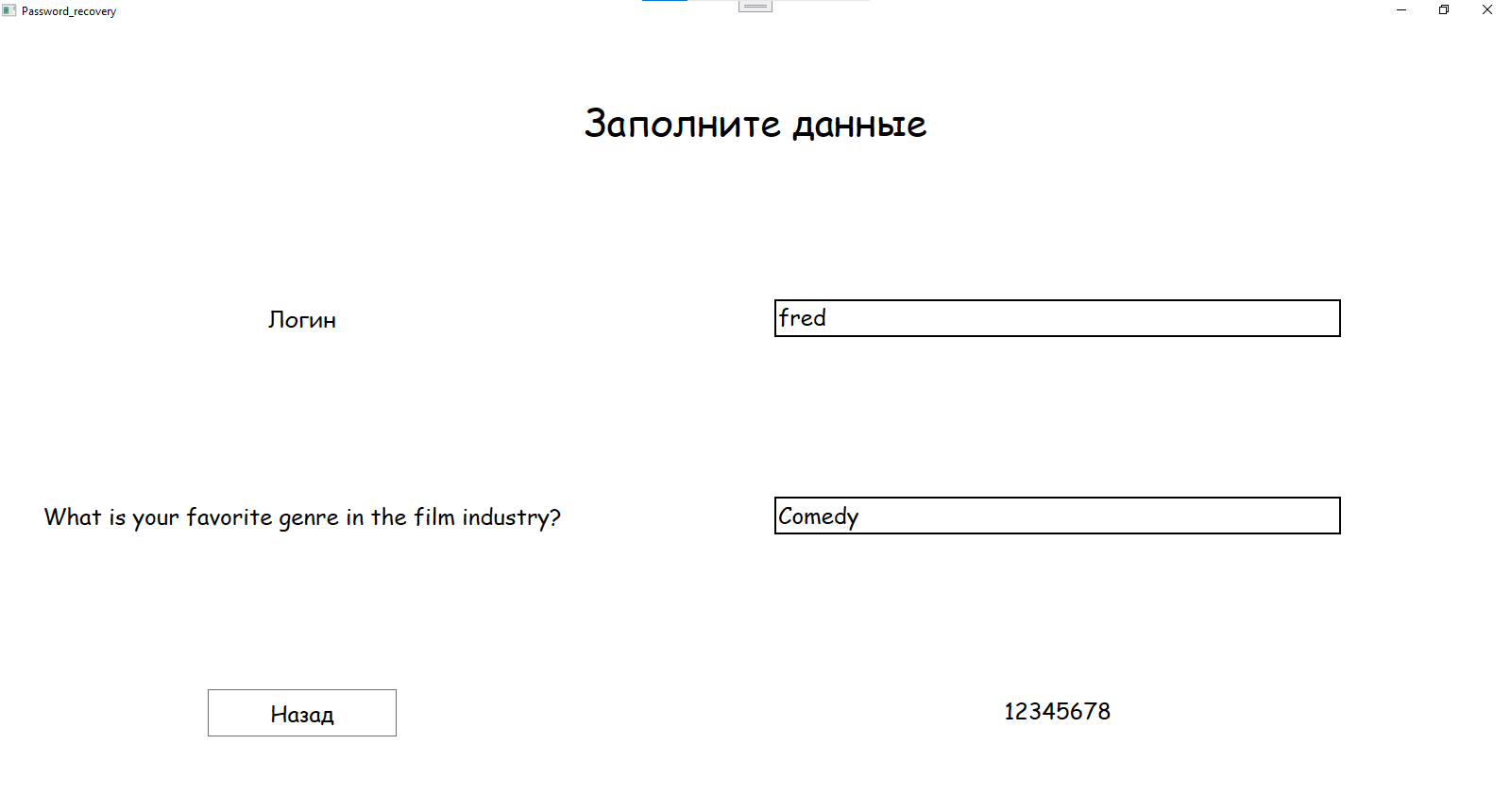


Рисунок 4 - Окно «Восстановление пароля»



Рисунок 5 - Окно профиля



Рисунок 6 - Детализация постов

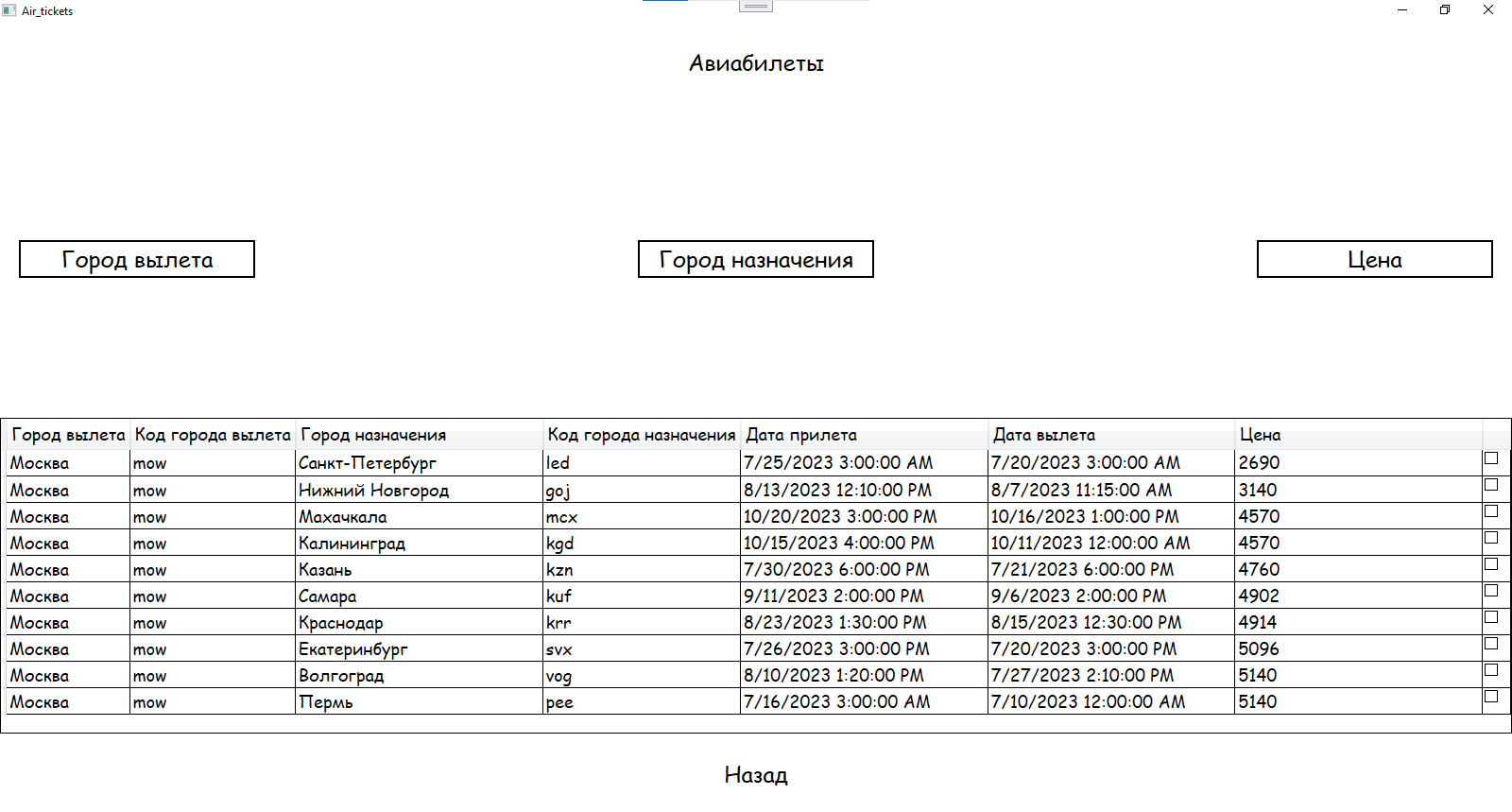


Рисунок 7 - Окно «Авиабилеты»

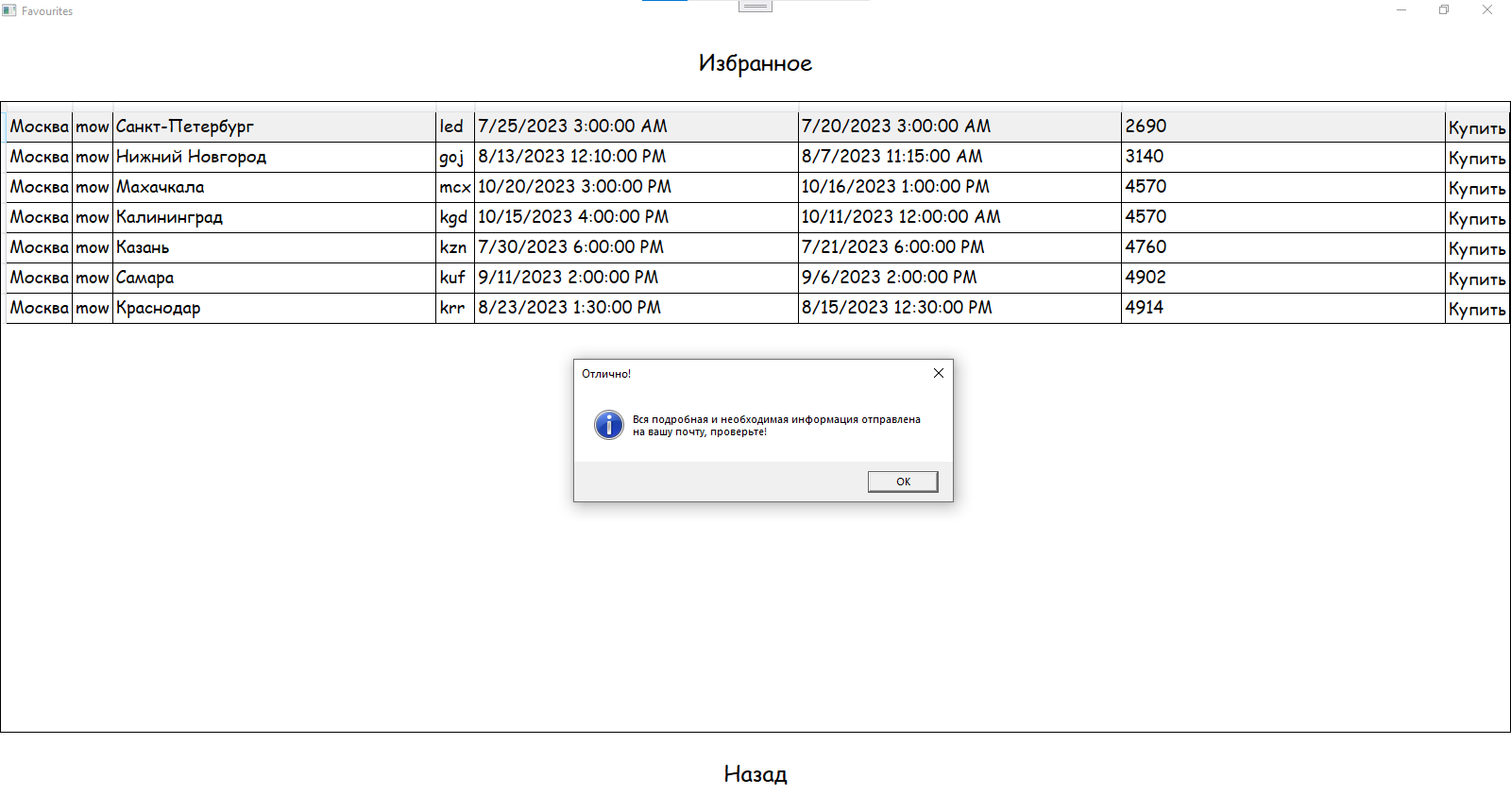


Рисунок 8 - Окно «Избранное»

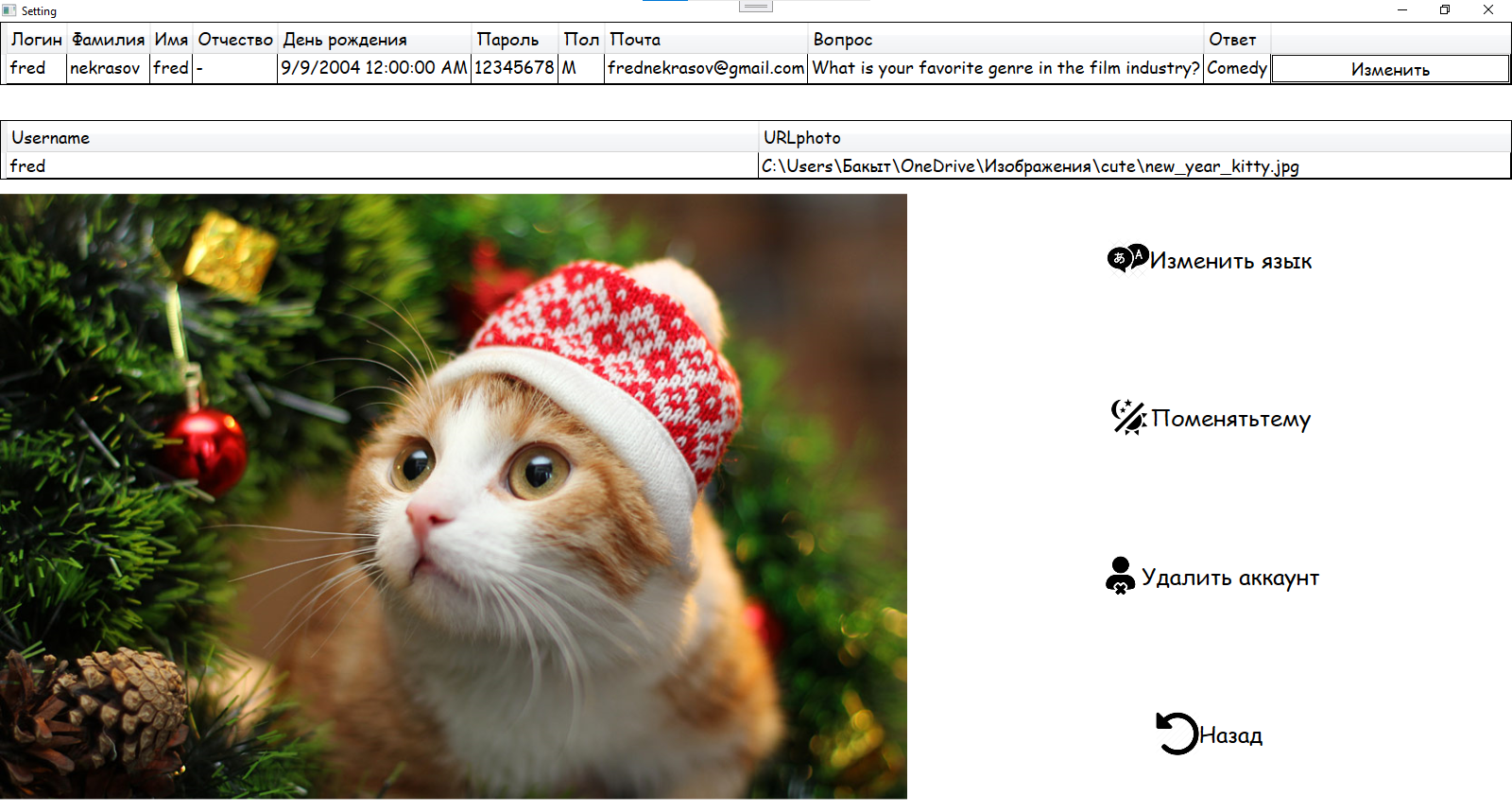


Рисунок 9 - Окно настроек

Ссылка на программу и тестирование <https://github.com/49fred99/Educational-practice.git>

# ПРИЛОЖЕНИЕ B

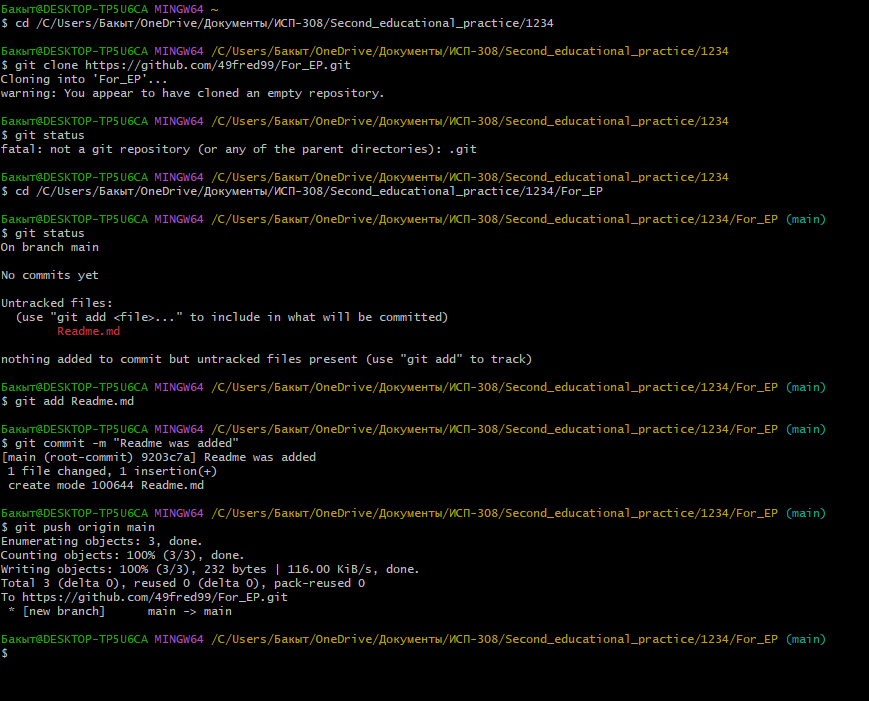


Рисунок 10 - Работа с базовыми командами git

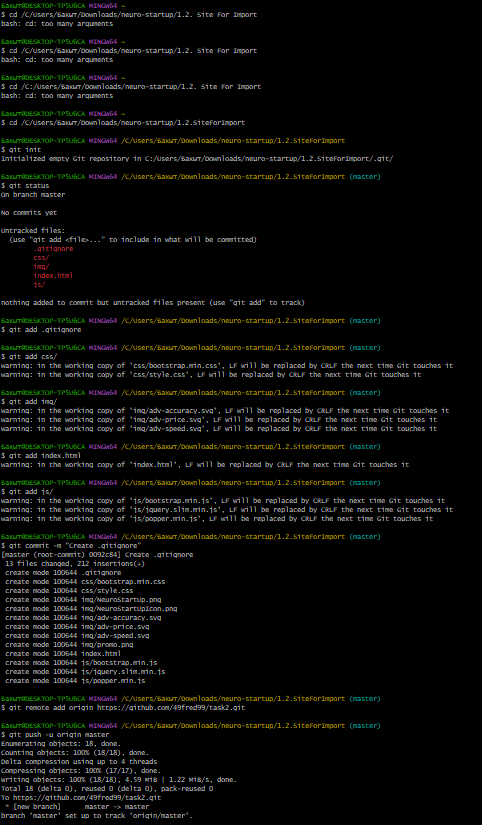


Рисунок 11 - Импорт существующего проекта



Рисунок 12 - Отмена изменений



Рисунок 13 - Создание веток

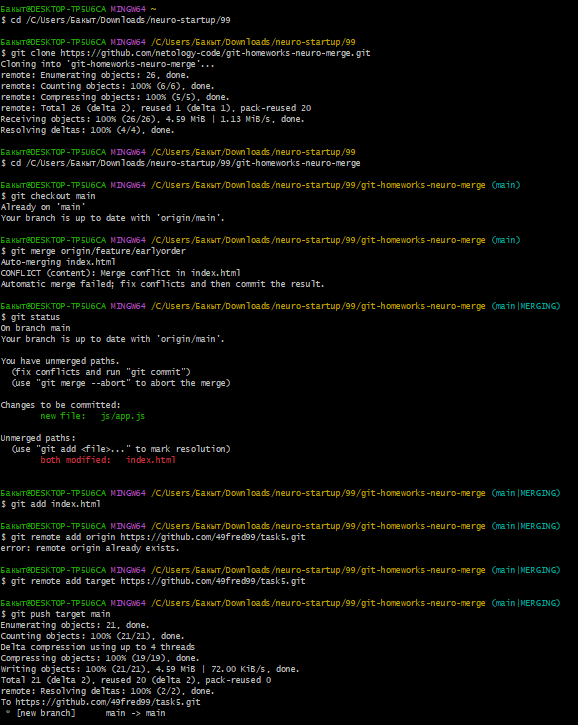


Рисунок 14 - Слияние изменений

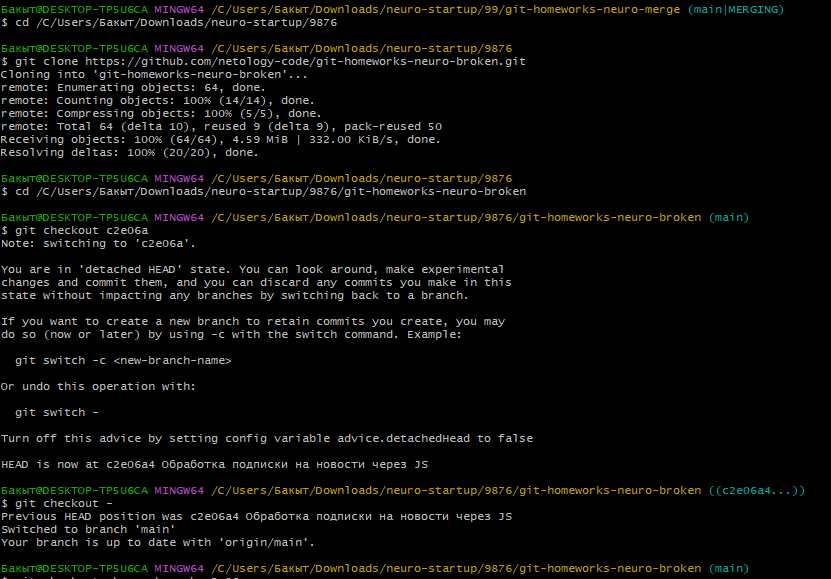


Рисунок 15.1 - Работа с историей и переключение между коммитами

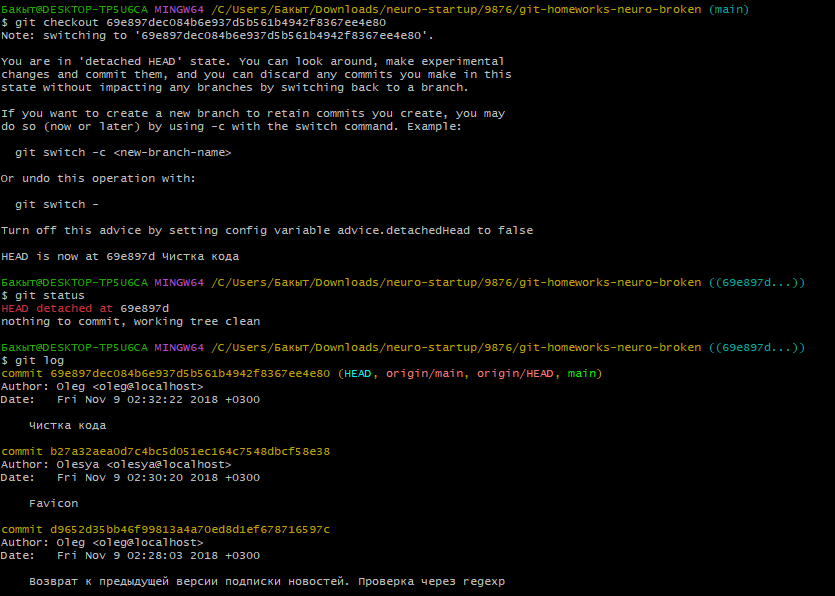


Рисунок 15.2 - Работа с историей и переключение между коммитами

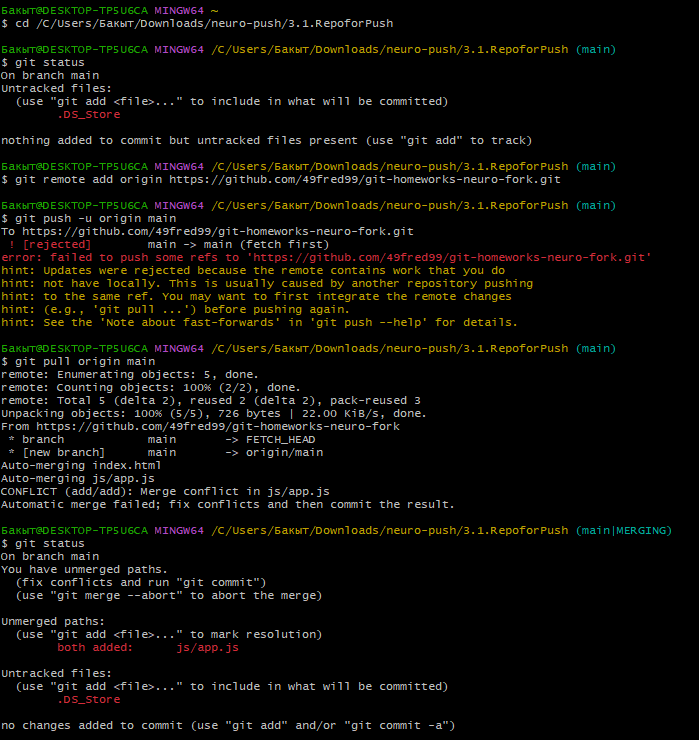


Рисунок 16.1 - Разрешение конфликта

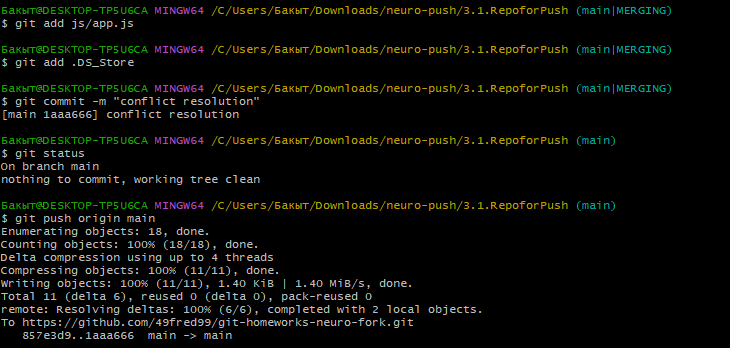


Рисунок 16.2 - Разрешение конфликта