**Паспорт проекта**

1. **Название проекта**

«Разработка мессенджера «SecureChat» на языке Python»

1. **Проектная группа**

Чвайковский Егор

1. **Аннотация**

Проект направлен на разработку мессенджера «SecureChat» с использованием языка программирования Python. Основная цель проекта — создание безопасного и удобного инструмента для обмена сообщениями с функцией шифрования данных. Мессенджер будет обеспечивать конфиденциальность общения благодаря встроенным механизмам криптографической защиты.

«SecureChat» включает в себя клиент-серверную архитектуру, где сервер обеспечивает управление учетными записями пользователей, обработку запросов и координацию передачи сообщений. Клиентская часть представляет собой интерфейс, через который пользователи смогут отправлять и получать зашифрованные сообщения.

Для реализации криптографических функций планируется использовать современные алгоритмы шифрования, чтобы обеспечить высокий уровень безопасности передаваемой информации.

Важной частью проекта является разработка интуитивно понятного и удобного графического интерфейса, что позволит пользователям легко ориентироваться в приложении и эффективно использовать его возможности.

Тестирование разработанного продукта проводится для проверки корректности работы всех компонентов системы, надежности защиты данных и устойчивости к различным видам атак.

Результатом проекта станет готовый к использованию мессенджер, предоставляющий пользователям безопасный канал связи с высоким уровнем конфиденциальности.

1. **Проблема**

Проблема, которую решает проект «SecureChat», заключается в отсутствии надежных и простых в использовании инструментов для защищенного обмена сообщениями. Современные мессенджеры часто не обеспечивают должный уровень конфиденциальности, что делает общение уязвимым для перехвата и несанкционированного доступа к личным данным. Это особенно актуально в условиях роста киберпреступности и утечек информации.

Мессенджер «SecureChat» направлен на решение этой проблемы путем внедрения современных методов шифрования, что позволяет гарантировать безопасность передаваемых сообщений. Проект предлагает удобный инструмент для повседневного использования, обеспечивающий защиту личной информации пользователей от посторонних лиц.

1. **Актуальность**

Актуальность проекта «SecureChat» обусловлена несколькими ключевыми факторами:

**Рост киберугроз**: В последние годы наблюдается значительный рост числа кибератак и случаев утечки личных данных. Пользователи все чаще сталкиваются с проблемами безопасности в интернете, что требует создания надежных инструментов для защиты их приватности.

**Необходимость в безопасной коммуникации**: Многие существующие мессенджеры либо не предлагают достаточно высокого уровня шифрования, либо делают процесс настройки безопасности сложным для обычных пользователей. «SecureChat» призван устранить этот пробел, предлагая простой и надежный способ защищенной коммуникации.

**Популяризация криптографии**: Проект способствует популяризации технологий шифрования среди широкой аудитории, демонстрируя, что защита персональных данных может быть доступна каждому.

**Образовательный аспект**: Разработка мессенджера на языке Python также имеет образовательную ценность, позволяя студентам и энтузиастам изучать принципы работы сетевых приложений и криптографии на практике.

Таким образом, «SecureChat» является актуальным проектом, направленным на удовлетворение растущих потребностей общества в защите личной информации и создании безопасных каналов связи.

1. **Цель проекта**

Цель нашего проекта заключается в разработке надежного и удобного мессенджера, обеспечивающего безопасное и конфиденциальное общение пользователей посредством шифрования передаваемых сообщений. Проект направлен на создание инструмента, который защитит личные данные пользователей от несанкционированного доступа и обеспечит высокую степень доверия к средствам коммуникации в сети Интернет.

1. **Задачи**

Для достижения цели проекта необходимо решить следующие задачи:

1. **Анализ требований и проектирование архитектуры приложения**:
   * Изучение существующих решений и технологий.
   * Определение основных модулей и компонентов мессенджера.
   * Проектирование клиент-серверной архитектуры.
2. **Разработка серверной части**:
   * Создание базы данных для хранения учетных записей пользователей.
   * Реализация механизмов аутентификации и авторизации.
   * Настройка серверного сокета для приема и передачи данных.
   * Интеграция алгоритмов шифрования для обеспечения безопасности передаваемых данных.
3. **Разработка клиентской части**:
   * Создание графического интерфейса пользователя (GUI) с использованием библиотеки Tkinter.
   * Реализация функционала отправки и получения сообщений.
   * Интеграция алгоритмов шифрования и дешифровки сообщений.
4. **Реализация механизмов шифрования**:
   * Выбор и внедрение алгоритмов шифрования (например, AES, RSA).
   * Разработка модуля для шифрования/дешифровки сообщений на стороне клиента.
   * Обеспечение синхронизации ключей шифрования между клиентами и сервером.
5. **Тестирование и отладка**:
   * Проведение функционального тестирования всех модулей приложения.
   * Нагрузочное тестирование для оценки производительности сервера.
   * Проверка корректности работы алгоритмов шифрования.
6. **Документирование и поддержка**:
   * Подготовка технической документации для разработчиков.
   * Создание руководства пользователя для клиентов.
   * Поддержка и обновление программного обеспечения после запуска.
7. **Оптимизация и масштабируемость**:
   * Оптимизация кода для повышения производительности.
   * Оценка возможностей масштабирования приложения для поддержки большого количества пользователей.

Эти задачи направлены на обеспечение надежной и эффективной работы мессенджера, соответствующего современным стандартам безопасности и удобства использования.

1. **Ожидаемые результаты**
2. **Готовый продукт:**
   * Создание полнофункционального мессенджера с поддержкой шифрования сообщений.
   * Приложение должно быть кросс-платформенным, работающим на различных операционных системах (Windows, macOS, Linux).
3. **Безопасность:**
   * Надежное шифрование передаваемых сообщений с использованием современных алгоритмов (AES, RSA).
   * Аутентификация и авторизация пользователей для предотвращения несанкционированного доступа.
4. **Удобство использования:**
   * Простой и интуитивно понятный графический интерфейс.
   * Возможность быстрой регистрации и начала общения.
5. **Масштабируемость:**
   * Способность поддерживать большое количество одновременных подключений.
   * Легкость расширения функциональности и добавления новых возможностей.
6. **Документация:**
   * Подробная техническая документация для разработчиков.
   * Руководства пользователя для эффективного использования мессенджера.
7. **Стабильность и надежность:**
   * Минимальное количество ошибок и сбоев в работе приложения.
   * Высокая производительность и скорость обработки данных.
8. **Образовательный эффект:**
   * Демонстрация принципов работы сетевых приложений и криптографии.
   * Возможности для студентов и энтузиастов изучить и улучшить приложение.
9. **Коммерческий потенциал:**
   * Потенциал для дальнейшего развития и монетизации проекта.
   * Привлечение инвесторов и партнеров для расширения бизнеса.
10. **Одержать победу на конкурсе проектов**

Эти результаты позволят создать востребованный и конкурентоспособный продукт, который удовлетворяет потребности пользователей в безопасном и удобном обмене сообщениями.

1. **Технические характеристики**

# Основные технологии и инструменты:

* **Язык программирования**: Python 3.x
* **Библиотека для графического интерфейса**: Tkinter (или альтернативные решения, такие как PyQt или Kivy)
* **Алгоритмы шифрования**:
  + Симметричное шифрование: AES (Advanced Encryption Standard)
  + Асимметричное шифрование: RSA (Rivest–Shamir–Adleman)
* **База данных**: SQLite (для локального хранения данных) или PostgreSQL / MySQL (для развертывания на сервере)
* **Сетевые протоколы**: TCP/IP для клиент-серверного взаимодействия
* **Формат передачи данных**: JSON или другой структурированный формат

# Архитектура:

* **Клиент-серверная архитектура**:
  + Серверная часть обрабатывает запросы клиентов, управляет базой данных и координирует передачу сообщений.
  + Клиентская часть реализует графический интерфейс и взаимодействие с пользователем.

# Ключевые компоненты:

* **Аутентификация и авторизация**:
  + Использование асимметричного шифрования (RSA) для безопасной передачи ключей.
  + Хэширование паролей перед сохранением в базу данных.
* **Шифрование сообщений**:
  + Передача сообщений происходит в зашифрованном виде с использованием симметричных ключей (AES).
  + Генерация уникальных сессионных ключей для каждого сеанса связи.
* **Синхронизация данных**:
  + Синхронизация состояния чатов и истории сообщений между клиентом и сервером.
* **Управление учетными записями**:
  + Регистрация новых пользователей.
  + Восстановление пароля через email или другие методы.

# Дополнительные функции:

* **Обмен файлами**: Возможность отправки файлов с применением тех же механизмов шифрования.
* **Групповые чаты**: Поддержка групповых бесед с безопасным обменом сообщений внутри группы.
* **Уведомления**: Оповещения о новых сообщениях и событиях в системе.

# Требования к инфраструктуре:

* **Серверная инфраструктура**:
  + Высокопроизводительный сервер для обработки большого объема запросов.
  + Резервное копирование данных для обеспечения отказоустойчивости.
* **Клиентские устройства**:
  + Поддержка различных операционных систем (Windows, macOS, Linux).
  + Минимальные требования к аппаратному обеспечению (процессор, оперативная память, место на диске).

1. **Сроки реализации**

Данный проект был выполнен за одну неделю.