Лабораторная работа 05

ПСКП

**Задание 01**

1. С помощью **npm** получите информацию о пакете **nodemailer**.

npm info nodemailer

1. С помощью **npm** скачайте пакет **nodemailer** в локальный директорий.
2. С помощью **npm** убедитесь, что пакет установлен.

**Задание 02**

1. Разработайте приложение **06-02**, использующее пакет **nodemailer.**
2. Приложение **06-02**, должно отправлять на браузер HTML-страницу, позволяющую ввести почтовые ящики отправителя и получателя, а также пересылаемое сообщение.
3. Убедитесь в работоспособности приложении.

**Задание 03**

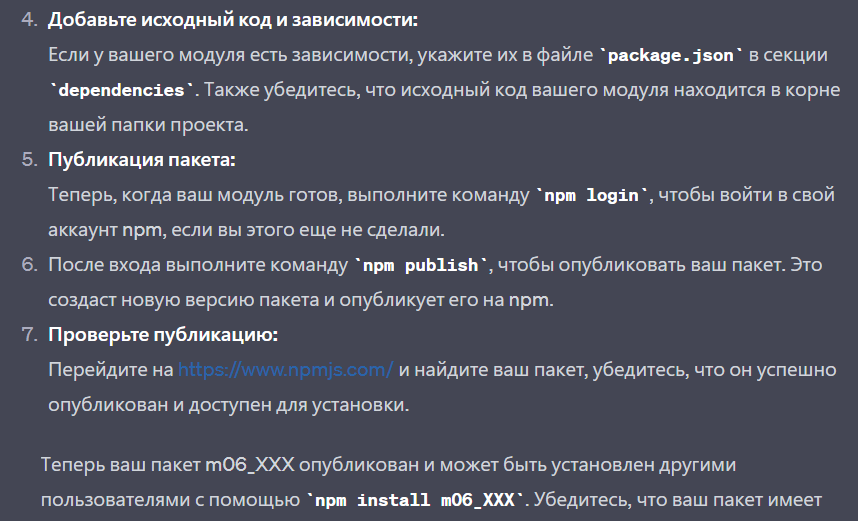
1. Разработайте модуль **m06\_XXX**, где XXX – ваши инициалы, который экспортирует одну функцию **send**.
2. Функция **send** принимает три параметра: почтовый адрес, на с которого будет отправлено сообщение, пароль и само сообщение.
3. Результатом выполнения функции **send**, является почтовое сообщение (email), содержащее строку, принятую в качестве параметра. Отправка сообщения осуществляется с помощью пакета **nodemailer**.
4. Сообщение, отправляемое функцией **send,** отправляется на указанный почтовый адрес.
5. Разработайте приложение **06-03,** применяющее модуль **m06\_XXX** и демонстрирующее работу функции **send** (примерный вид приложения представлен на сл. рис).



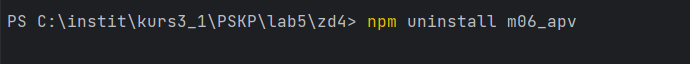
1. Убедитесь в работоспособности приложения.

**Задание 04**

1. Опубликуйте пакет **m06\_XXX**, содержащий соответствующий модуль.
2. Зайдите на сайт <http://npmjs.com> и убедитесь, что пакет опубликован.



1. Скачайте пакет в локальный репозиторий и разработайте приложение **06-04,** демонстрирующее применение скачанного пакета.
2. 
3. Удалите пакет **m06\_XXX** из локального репозитория.



1. Установите пакет **m06\_XXX** в глобальный репозиторий и проверьте работоспособность приложения **06-04.**





**Задание 05**

1. Поясните назначение **npm**.

**npm (Node Package Manager)** - это менеджер пакетов, который используется в экосистеме Node.js для установки, управления и обновления пакетов JavaScript. Он позволяет разработчикам легко интегрировать сторонние библиотеки и фреймворки в свои проекты, а также делиться своими собственными пакетами с сообществом.

1. Перечислите команды **npm**,с помощью которых можно просмотреть список установленных пакетов.
2. **npm list** или **npm ls**: Показывает список установленных пакетов и их зависимостей.
3. **npm list -g**: Показывает список глобально установленных пакетов и их зависимостей.
4. Поясните назначение параметра **–g** в командах **npm.**
5. Параметр **-g** в командах **npm** означает глобальную установку пакета. Глобальная установка делает пакет доступным системно на уровне пользователя, что позволяет использовать пакет в разных проектах и даже вне проектов. Например, **npm install -g package-name** установит пакет глобально.
6. Перечислите команды **npm**, с помощью которых можно скачать пакет.

* **npm install package-name**: Устанавливает пакет локально в текущем проекте.
* **npm install -g package-name**: Устанавливает пакет глобально.
* **npm install -D package-name** или **npm install --save-dev package-name**: Устанавливает пакет в качестве зависимости разработки в текущем проекте и добавляет его в секцию **devDependencies** файла **package.json**.

1. Поясните назначение файла **package.json**.

**package.json** - это файл конфигурации проекта Node.js. Он содержит метаданные о проекте, включая его имя, версию, зависимости и скрипты для управления проектом. Этот файл также используется для определения списка зависимостей, которые должны быть установлены в проекте.

1. Перечислите последовательность действий, позволяющих опубликовать пакет с помощью **npm**.
2. **Создайте аккаунт на npm (если его нет)**: Перейдите на [сайт npm](https://www.npmjs.com/) и создайте аккаунт, если у вас его еще нет.
3. **Логин в npm**: В терминале выполните команду **npm login** и введите учетные данные своего аккаунта npm.
4. **Подготовьте свой пакет**: Убедитесь, что ваш пакет имеет файл **package.json**, содержащий необходимую информацию о пакете.
5. **Увеличьте версию пакета (если необходимо)**: Используйте команду **npm version** или редактируйте версию вручную в файле **package.json**. Версия должна быть уникальной.
6. **Опубликуйте пакет**: Выполните команду **npm publish** в корневой директории вашего проекта. Пакет будет опубликован на npm.
7. **Обновите пакет (по желанию)**: Если вам нужно выпустить новую версию пакета, повторите шаги 4 и 5 с новой версией.