**Министр науки и высшего образования Российской̆ Федерации**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Национальный исследовательский университет ИТМО»**

Факультет информационных технологий и программирования

Лабораторная работа №3

*Разработка плагина для IDE*

**Выполнил студент группы № M3113**

Ващенко Тимур Денисович

**Подпись:**

**Проверил:**

Трусилов Артем Алексеевич

Санкт-Петербург

2025

1. Зачем я это делал

Хотел понять, как работают расширения в VS Code. Обычный таймер в отдельном окне — мусор. А если засунуть его в статус-бар (полоску внизу VS Code) — будет всегда видно, не отвлекает. Вот и сделал.

1. Как устроены расширения VS Code (на пальцах)

VS Code — это приложение. Но расширения работают в отдельном процессе, как отдельная программка.

Когда ты устанавливаешь расширение:

VS Code читает файл package.json (там описано, что это за расширение)

Запускает функцию activate() — расширение "просыпается"

Расширение может слушать события, создавать кнопки, окна

Когда ты закрываешь VS Code, вызывает deactivate() — расширение выключается

Это как маленькая программка, которая живёт внутри VS Code.

1. Что я сделал в наше Pomodoro

Две части UI:

1. Статус-бар (полоска внизу справа)

Вот это видно всегда: 🍅 Work 25:00 ●

Кликаешь — включается/выключается таймер

Иконка меняется: помидор для работы, кофе для перерыва

Кружочек: ● если идёт отсчёт, ○ если паузу

2. Контрол-панель (красивое окно)

Открывается отдельно по команде

Большой таймер в центре (80px шрифт)

Кнопки: Старт, Стоп, Ресет

Слайдеры для настройки времени работы/перерыва

Переключатели для звука и уведомлений

Считает сколько сессий закончилось

Как это на самом деле работает

1. Таймер (главная логика)

У расширения есть переменные:

timeLeft — сколько секунд осталось

isRunning — идёт ли отсчёт

isWorkSession — сейчас работа или перерыв

Каждые 100 миллисекунд расширение проверяет: "сколько реального времени прошло?" и вычитает из счётчика. Так что если ноутбук зависнет на секунду — таймер не сбивается, он считает по реальному времени.

Когда время кончается:

Звук (простенький beep через Web Audio)

Уведомление: "🍅 Время работы закончилось!"

Переключается на перерыв

После 4 сессий работы — длинный перерыв 15 минут вместо 5

Как статус-бар и панель общаются

Они в разных процессах. Статус-бар живёт в расширении, панель — это вебстраница (как отдельный браузер).

Общаются через сообщения:

Расширение каждую секунду шлёт панели: "вот новое время"

Панель обновляет цифры

Когда ты кликаешь кнопку в панели — она отправляет сообщение расширению

Расширение получает команду и выполняет действие

Это как двусторонний чат, но между двумя частями программы.

Настройки

Ты двигаешь слайдер "Work Duration" в панели:

Панель отправляет новое значение (например, 30 минут)

Расширение получает, сохраняет в конфиг VS Code

VS Code сохраняет в файл на диске

Даже если перезагрузишь — настройки остаются

Это как браузер сохраняет пароли, но для VS Code.

1. Что я узнал

Архитектура:

Расширения — это отдельный Node.js процесс

VS Code общается с ними через API

Есть встроенные UI элементы (статус-бар, команды, уведомления)

Синхронизация:

Две части расширения (статус-бар и панель) живут в разных процессах

Общаются через очередь сообщений

Нужно держать их в синхе (отправлять обновления туда-сюда)

UI в VS Code:

StatusBar — простой текст справа внизу, можно кликнуть

Webview — это как браузер внутри VS Code, там можно HTML/CSS/JavaScript

Уведомления — всплывающие сообщения

Точность таймера:

Нельзя просто делать -1 каждую секунду

Нужно считать по реальному времени (Date.now())

Иначе при лагах и нагрузке таймер сбивается

Как я это запускал

Создал папку с файлами

npm install (скачал нужные пакеты)

Открыл в VS Code

Нажал F5 → открылось второе окно VS Code с активным расширением

Кликнул на 🍅 в статус-баре → таймер запустился

Команда "Show Control Panel" → открылась панель с настройками

Что в итоге получилось

Минимальный рабочий Pomodoro, который:

Сидит в статус-баре и не мешает

Красивый интерфейс в отдельной панели

Можно настраивать прямо в приложении

Работает с точностью (не сбивается при лагах)

Настройки сохраняются

И самое главное — я разобрался с тем, как устроены расширения VS Code. Это не какая-то магия, это обычный Node.js код, который общается с VS Code через API.