Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления
Кафедра информационных технологий автоматизированных систем

Отчёт по лабораторной работе №2
Вариант 5

по дисциплине
«Интернет-технологии и веб-программирование»

НАСТРОЙКА РАБОЧЕГО ОКРУЖЕНИЯ ДЛЯ ВЕБ-РАЗРАБОТКИ

Выполнил: студ. гр 320604
Колодич А.Д.

Проверила: Хаджинова Н.В.

Минск 2026

СОДЕРЖАНИЕ

1 ЦЕЛЬ РАБОТЫ

Формирование практических навыков настройки современного рабочего окружения для полноценной веб-разработки. Освоение инструментов контроля версий, создание и конфигурация проекта, изучение базовых возможностей интегрированной среды разработки и инструментов проектирования.

2 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Процесс веб-разработки — это комплекс действий, направленных на создание и поддержку сайтов и веб-приложений. Он начинается с анализа требований, где определяются цели проекта, его функциональные возможности и целевая аудитория. Затем выполняется проектирование: разрабатывается структура сайта, создаются прототипы страниц и продумывается удобство взаимодействия пользователя с интерфейсом.

Основной этап — разработка, которая делится на *frontend* и *backend*. *Frontend* отвечает за внешний вид сайта и реализуется с помощью *HTML*, *CSS* и *JavaScript*. *Backend* обеспечивает серверную логику, обработку данных и взаимодействие с базами данных. После программирования проводится тестирование для выявления ошибок и проверки корректной работы сайта на разных устройствах и в браузерах. Завершающими стадиями являются развертывание проекта на сервере и его дальнейшая поддержка.

Системы контроля версий используются для отслеживания изменений в коде и организации командной работы. Наиболее популярной системой является *Git*, позволяющая сохранять историю проекта, работать с ветками и при необходимости возвращаться к предыдущим версиям. Для хранения репозиториев часто применяются платформы *GitHub* и *GitLab*, которые также предоставляют инструменты для совместной разработки.

Инструменты разработчика помогают ускорить и упростить процесс создания веб-приложений. К ним относятся редакторы кода и среды разработки, например *Visual Studio Code*, которые предлагают подсветку синтаксиса и средства отладки. Встроенные инструменты браузеров позволяют анализировать структуру страниц и находить ошибки, а дополнительные технологии, такие как менеджеры пакетов и сборщики проектов, автоматизируют многие задачи разработки.

3 ХОД РАБОТЫ

В начале выполнения лабораторной работы было установлено необходимое программное обеспечение для организации рабочего окружения веб-разработчика. С официального сайта была загружена *LTS* версия *Node.js*, после установки в терминале *Visual Studio Code* выполнена проверка корректности работы с помощью команд *node --version* и *npm --version*. Полученные значения подтвердили успешную установку платформы и менеджера пакетов. Результат представлен на рисунке 1.

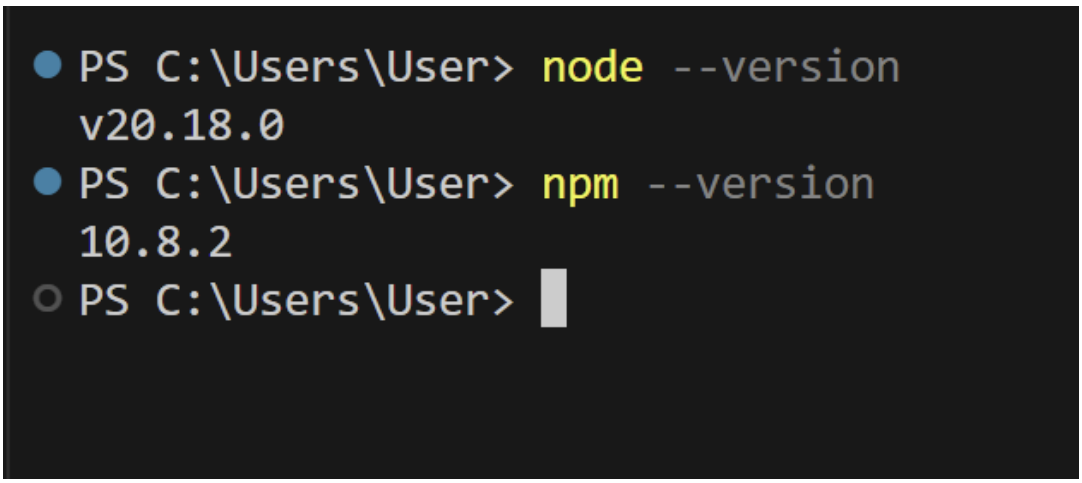

Рисунок 1 – Проверка версий *Node.js* и *npm*

Далее была выполнена установка системы контроля версий *Git*. После завершения установки в терминале проверена доступность команды *git --version*, что подтвердило готовность инструмента к работе. Затем были настроены глобальные параметры пользователя, включающие имя и адрес электронной почты, необходимые для корректного ведения истории изменений проекта. Результат представлен на рисунке 2.

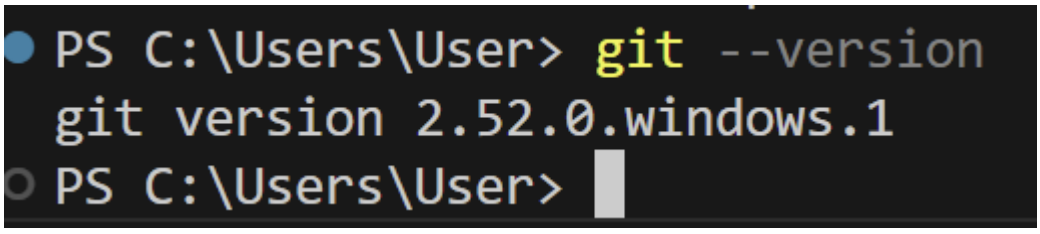

Рисунок 2 – Проверка версии *Git*

Следующим этапом стала установка интегрированной среды разработки *Visual Studio Code*. После запуска программы был изучен интерфейс среды, включая панель активности, редактор кода, боковую панель, строку состояния и встроенный терминал. Для повышения эффективности разработки были установлены расширения *Live Server, Auto Rename Tag, Prettier, GitLens, ES6 String HTML* и *Figma to Code*. Результат представлен на рисунке 3.

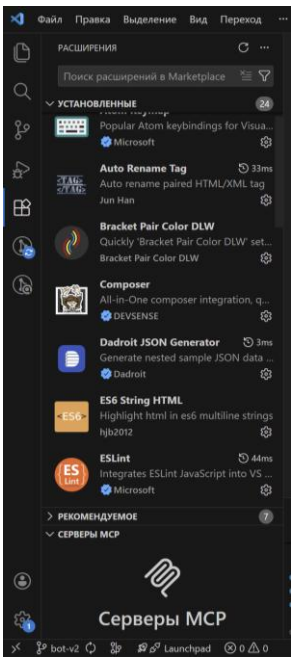

Рисунок 3 – Установленные расширения в *VS Code*

После подготовки среды был создан новый каталог проекта и открыт в *Visual Studio Code*. Через встроенный терминал сформирована структура папок проекта, предназначенных для хранения таблиц стилей, скриптов, изображений и дополнительных ресурсов. Далее были созданы базовые файлы *index.html*, *style.css* и *script.js*, которые обеспечивают основу для дальнейшей разработки веб-приложения.

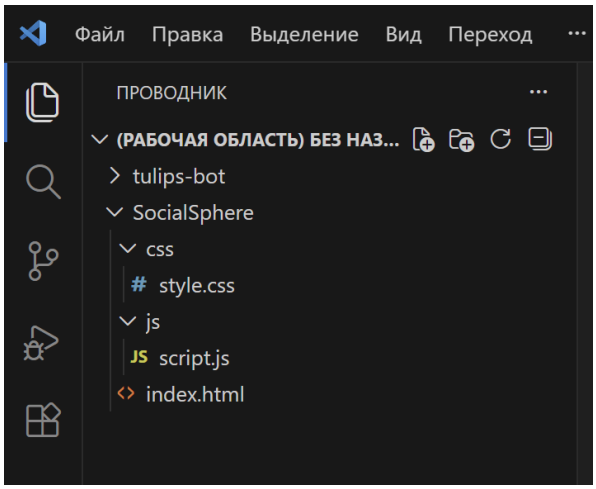

Рисунок 4 – Структура проекта

На следующем этапе состоялось знакомство с инструментом проектирования *Figma*. Был создан аккаунт и новый файл проекта. В рабочем пространстве разработан мудборд, определена цветовая палитра и выбрана типографика будущего сайта. Также был подготовлен простой прототип главной страницы, отражающий структуру интерфейса.

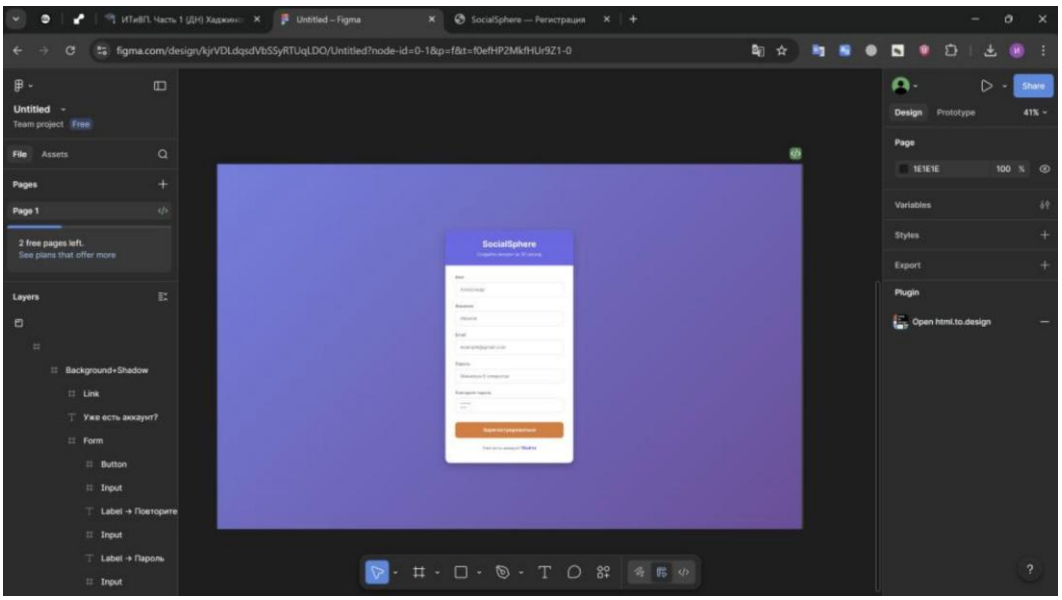

Рисунок 5 – Прототип страницы в *Figma*

После этого выполнена инициализация *Git*-репозитория в корневой папке проекта. Создан файл *gitignore* для исключения служебных и временных файлов из отслеживания. С помощью вкладки *Source Control* файлы были добавлены в область индексирования и выполнен первый коммит с начальной структурой проекта.

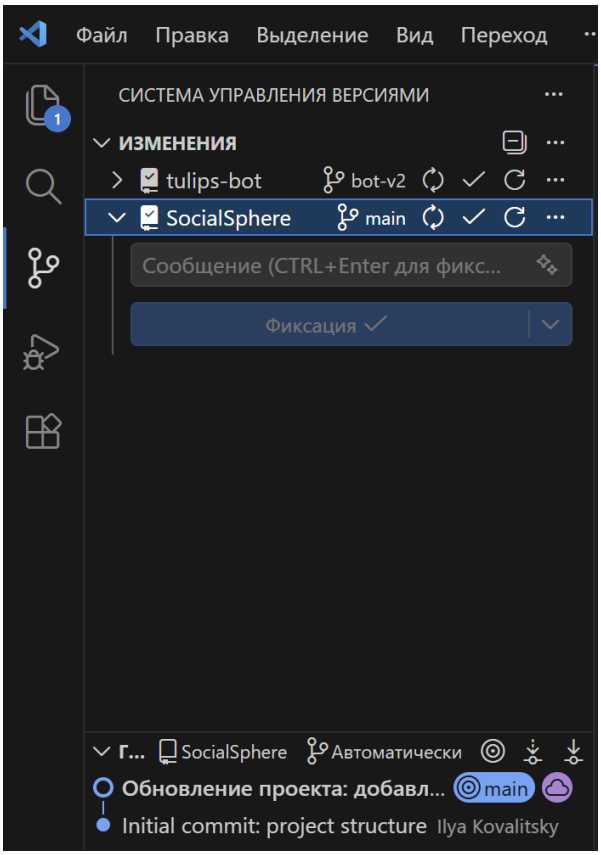

Рисунок 6 – История коммитов

Далее создан удаленный репозиторий на платформе *GitHub*, после чего локальный проект был связан с удаленным хранилищем и отправлен на сервер. Это позволило обеспечить резервное хранение проекта и возможность совместной разработки.

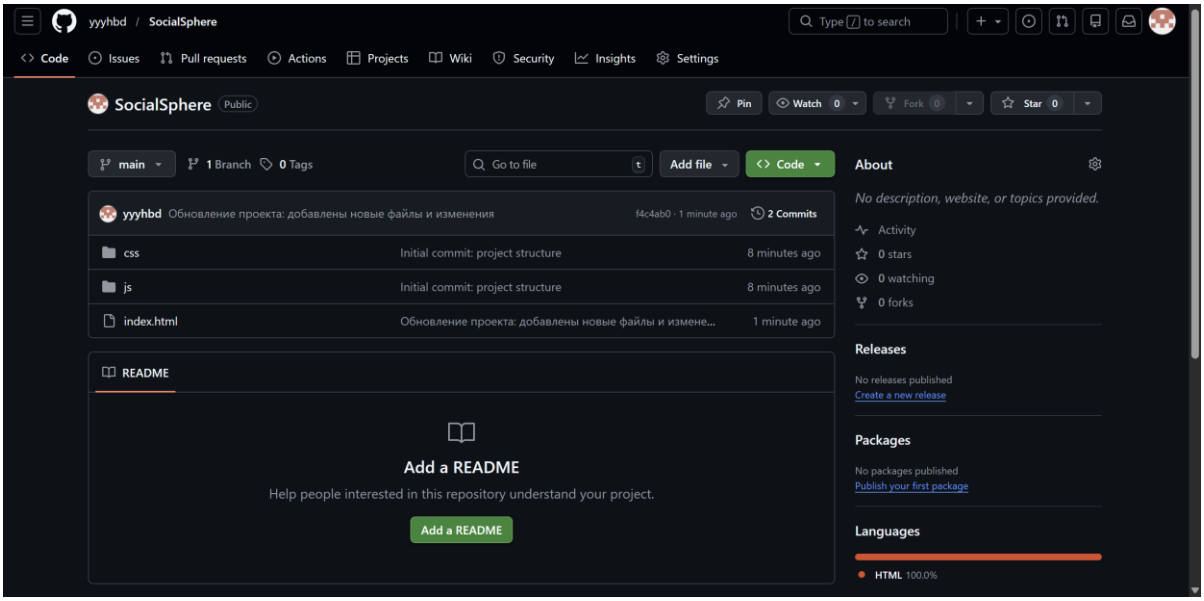

Рисунок 7 – Удаленный репозиторий проекта

Затем был инициализирован *npm* проект с автоматическим созданием файла конфигурации. Установлены необходимые зависимости, а также инструмент для разработки, упрощающий запуск приложения. Для поддержания единого стиля кода создан файл конфигурации форматирования и настроено автоматическое форматирование при сохранении файлов.

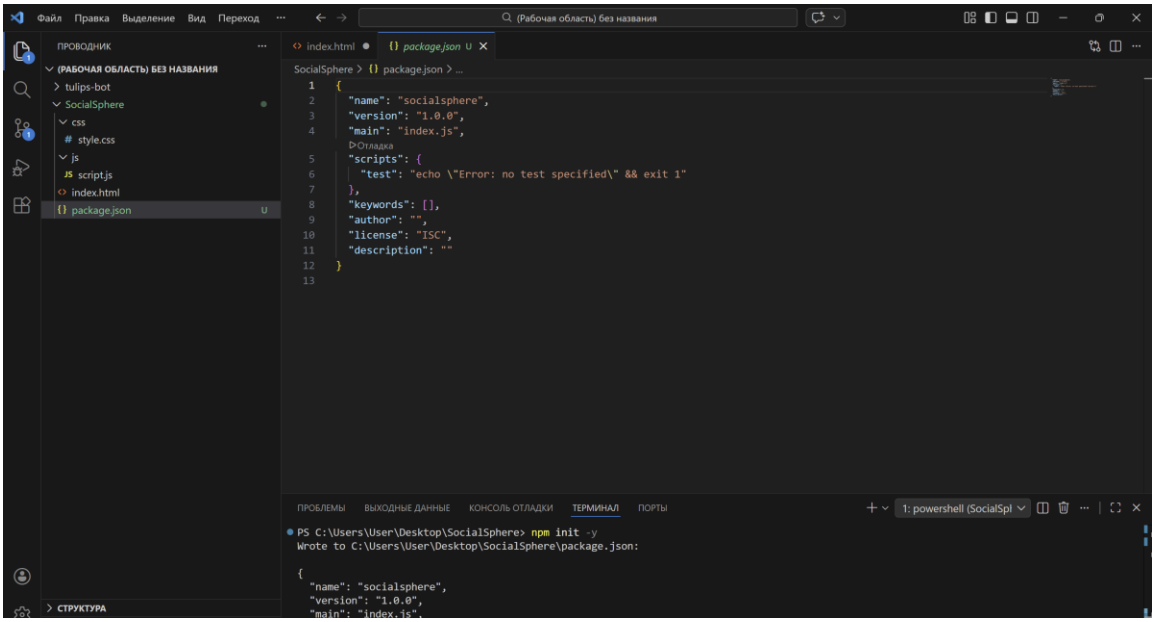

Рисунок 8 – Инициализация *npm* проекта

Ниже приведены ответы на 2 контрольных вопроса:

1 *Node.js* и *npm* играют ключевую роль в управлении зависимостями проекта, позволяя через пакетный менеджер автоматизировать установку библиотек, гарантировать совместимость версий используемых модулей и значительно ускорять процесс сборки веб-приложения;

2 Инструменты проектирования *Figma* интегрируются в процесс разработки как источник визуальных спецификаций, позволяя на этапе подготовки к верстке точно извлекать *CSS*-свойства элементов, параметры типографики и экспортировать графические ресурсы для их переноса в код.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы были сформированы практические навыки настройки современного рабочего окружения для полноценной веб-разработки. Освоены ключевые этапы инициализации проекта: развертывание *npm*-среды с автоматической генерацией конфигурационных файлов, установка и интеграция необходимых зависимостей, а также подключение инструментов, обеспечивающих удобство разработки и автоматизацию рутинных процессов.