

граммы Adobe Photoshop. Вся необходимая информация уже имеется, это цветовая гамма и схематические прототипы имеющихся страниц;

– верстка страниц и программирование. На данном этапе приступаем к верстке сайта магазина «Окна-Двери». Переводим дизайн сайта в html вид аналогичный макету, используя при этом CSS, HTML и CMS-ModX.

– на данном этапе создаем базу данных, подключаем и настраиваем все компоненты на хостинг, проводим тестирование созданного сайта.

Это пять основных этапов, при разработке сайта магазина «Окна-Двери». По окончании последнего этапа программирования сайта получаем полностью готовый к использованию интернет-магазин.

А.С. Чеботаревский (ГГУ имени Ф. Скорины, Гомель),

Е.А. Левчук (БТЭУ ПК, Гомель),

Науч. рук. **Е.А. Левчук**, канд. техн. наук, доцент

ПАРСИНГ SPA С ПОМОЩЬЮ GOOGLE PUPPETEER

При парсинге статических сайтов разработчик может забирать всю страницу и с помощью регулярных выражений выбирать данные, которые ему требуются. Если требуется получить данные с веб ресурса, созданного как Single Page Application (SPA), подобный подход становится невозможным, так как данные подгружаются уже после загрузки страницы, могут обновляться после определенных действий пользователя.

Чаще всего при попытке получить данные с SPA с помощью классических статических методов разработчик получает практически полностью пустую HTML страницу. Также в некоторых случаях получить данные можно, лишь совершив некоторый набор действий, которые может выполнить обычный пользователь.

Для парсинга подобных веб ресурсов используется программное обеспечение, которое запускает браузер и эмулирует действия пользователя. Если запускать полновесный браузер, то каждая отдельная попытка получить данные с SPA будет тратить большое количество вычислительных мощностей, поэтому на данный момент для решения задач по парсингу SPA используются так называемые headless браузеры, которые являют собой полноценный браузер без графической оболочки. Также не стоит забывать, что разные браузеры могут выводить информацию по-разному, а на некоторых браузерах современ-

ные сайты не будут работать. Таким образом, наиболее верным решением будет использовать в качестве headless браузера самый популярный браузер в мире - Google Chrome.

На данный момент существует библиотека Puppeteer, которая имеет очень удобный API с большой документацией по управлению headless Chrome browser. Для работы с Puppeteer требуются установленные утилиты node js и npm. В этом случае с помощью библиотеки можно имитировать такие действия пользователя, как движения мышкой, нажатия кнопок, ввод данных в формы ввода, ожидания и задержки. Puppeteer имеет наиболее полную документацию и широкую поддержку, что делает ее одним из лучших плагинов по работе с headless браузерами.

Примеры использования библиотеки Puppeteer для обработки информации из различных источников обсуждаются в докладе.

А.В. Черенко (ГГУ имени Ф. Скорины, Гомель)
Науч. рук. **А.И. Кучеров**, ст. преподаватель

АРХИТЕКТУРА КОМПЛЕКСА ПО ПРЕДОТВРАЩЕНИЮ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА К УЗЛУ ЛВС

В текущий момент информация имеет большую ценность, поэтому первоочередной задачей является обеспечение безопасности данных. Стойкость подсистемы аутентификации определяется гарантией того, что злоумышленник не сможет пройти процедуру подтверждения личности. Для повышения стойкости, помимо классических систем аутентификации, можно использовать технологию мониторинга рабочей нагрузки, порождаемой пользователем. А также процедуры проверки пользователя на соответствие с заранее составленным портретом.

Программный комплекс по получению информации о используемых приложениях и времени их работы в операционной системе будет снимать различные показатели с вычислительной системы. Исходя из этих данных можно будет судить о характере рабочей нагрузки создаваемой пользователем в вычислительной системе, а также классифицировать рабочую нагрузку.

В настоящее время существуют решения, позволяющие отслеживать рабочую нагрузку, однако данные решения не способны прини-