

Гордеев Д.А.

аспирант,

dan.gor220@yandex.ru

Московский финансово-юридический университет,

г. Москва

ФОРМИРОВАНИЕ ПРАКТИЧЕСКИХ НАВЫКОВ У УЧАЩИХСЯ ТЕХНИЧЕСКИХ НАПРАВЛЕНИЙ ПРИ ИЗУЧЕНИИ ВЕБ-ПРОГРАММИРОВАНИЯ

В статье предложен подход к обучению веб-программированию, основанный на практическом применении изученного материала, проектной работе в команде и взаимном оценивании проектных работ. Целью исследования является разработка курса по формированию практических навыков у учащихся технических направлений при изучении веб-программирования. Недостаточная разработанность методических основ по формированию практических компетенций веб-программирования говорит об актуальности исследования. При исследовании были задействованы методы анализа, проектной деятельности и взаимной оценки проектной деятельности. В результате исследования систематизированы теоретические разделы по ряду веб-технологий, выбрана платформа для алгоритмического развития навыков учащихся, разработано веб-приложение, анализирующее успеваемость по выполнению алгоритмических задач, разработаны учебные проекты и техническая спецификация к ним, сформулированы критерии оценивания проектных работ и описаны этапы семинарских занятий по взаимному оцениванию.

Ключевые слова: *Веб-программирование, проектная деятельность, взаимное оценивание, алгоритмы, javascript, git, github, api*

Развитие современного общества характеризуется, в частности, активным применением информационных технологий во всех сферах деятельности. Прогресс информационных технологий, необходимость поддержки информационных продуктов и ресурсов на уровне современных мировых тенденций определяет потребность в подготовке высококвалифицированных ИТ-специалистов [1].

Традиционные программы обучения предполагают наличие у учащегося прerreквизитов в области информационных технологий, начальных

навыков в программировании, полученных в результате изучения информатики в общеобразовательных учреждениях, а также содержат подходы, несоответствующие современным требованиям к развитию навыков у кандидата на позицию веб-разработчика, что приводит к необходимости переобучения к текущим требуемым реалиям и, впоследствии, к возможному отсутствию дальнейшей мотивации на обучение [2, 3, 4, 5]. Разработанный курс носит практико-ориентированный характер и для устранения возможных пробелов в базовых знаниях разработаны теоретические разделы с элементами практики по ряду веб-технологий, изучение которых позволит учащемуся успешно освоить необходимые знания.

Первый этап практической части курса заключается в выполнении ряда алгоритмических задач с автоматическим тестированием. В качестве платформы для решения задач выбрано образовательное сообщество по компьютерному программированию «codewars». Решение задач позволит начинающему разработчику развить алгоритмическое мышление и закрепить знание основ синтаксиса языка программирования «JavaScript» [6, 7].

Для автоматизации проверки успеваемости учащихся в выполнении алгоритмических задач разработано веб-приложение, которое по имени пользователя на платформе «codewars» определяет выполнена задача учащимся или нет. Веб-приложение реализовано с помощью API-интерфейса платформы «codewars». При разработке приложения использовался планировщик задач «gulp», сборка которого позволила реализовать веб-приложение кросс-браузерным, кроссплатформенным и минифицированным [7]. Структура проекта представлена на рисунке 1.

Левая часть схемы отображает зависимости для работы с планировщиком задач «gulp» и npm-пакетами. С правой стороны расположены две папки, «dist» – место куда компилируются конечные файлы и «src» – папка для разработки проекта.

Для проверки выполнил учащийся задание или нет используется обработчик событий, который срабатывает при нажатии на кнопку «check». Логика обработчика заключается в получении строкового значения имени пользователя из поля ввода. После чего происходит запрос на API-платформу «codewars» и возвращается массив данных о выполненных задачах

пользователя. Если в возвращаемых данных присутствует идентификатор задания из ранее определенного массива «arrID», то к элементу «li» добавляется класс «completed», в обратном случае, добавляется класс «notCompleted». В результате приложение визуально отображает статус готовности задачи (рисунок 2).

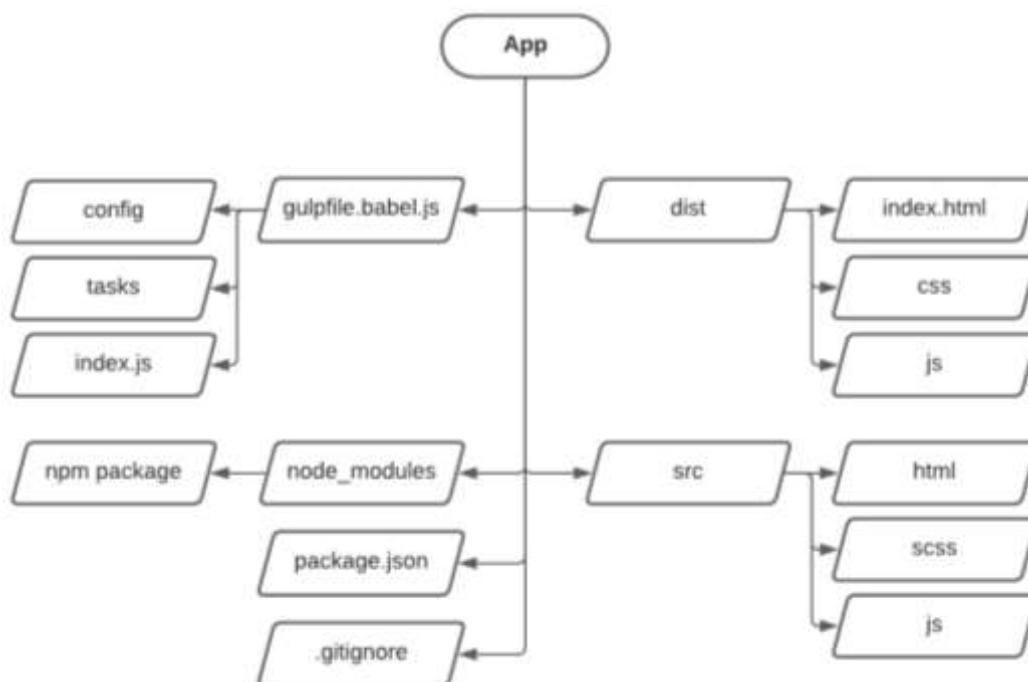


Рисунок 1 – Схема проекта веб-приложения

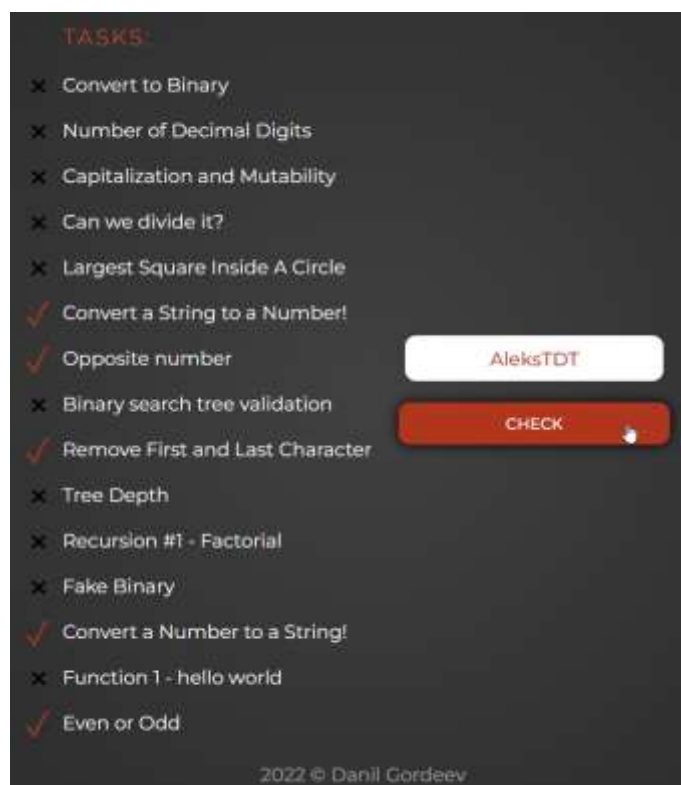


Рисунок 2 – Работа веб-приложения

После выполнения и проверки алгоритмических задач наступает этап разработки проектов. На данном этапе не только оттачиваются навыки в разработке, но и формируется «портфель» с работами, который впоследствии можно показать работодателю [8].

Первый проект разработан с целью создания индивидуальной веб-страницы учащегося, на которой будут отображены практические навыки, полученные в ходе прохождения курса, примеры кода, написанного в ходе решения алгоритмических задач, командный проект и последующие проекты, которые будущий разработчик сможет добавлять в разработанное портфолио.

Перед разработкой проекта учащемуся необходимо ознакомиться с критериями оценивания проекта, требованиями к проекту и верстке, после чего приступить к выполнению. После выполнения проекта, учащийся загружает ссылку на репозиторий с выполненным проектом, в раздел курса «взаимное оценивание проекта "Портфолио"».

Современная веб-разработка также предполагает наличие команды, каждый член которой выполняет определенную задачу [5]. Работа в команде ключевой навык, которым должен обладать веб-разработчик. Для подготовки учащегося к командной разработке реализован второй проект, в результате которого группы учащихся спроектируют и разработают многостраничное веб-приложение, используя знания, полученные в ходе выполнения курса. Главный упор проекта рассчитан на совместную работу учащихся, разделенных на команды из 3-4 человек, и применения системы контроля версий Git.

Порядок выполнения проекта:

- изучить требования к проекту;
- разработать базовую структуру проекта;
- определиться с обязанностями, функционалом проекта, подлежащим выполнению;
- командир команды создает публичный «GitHub» репозиторий, после чего от ветки «master» создает ветку «dev». Ветка «master» – это основная «production» ветка, в которой будет финальная версия проекта. Вся разработка ведется в ветке «dev»;

- все участники команды клонируют репозиторий в локальную рабочую область, переходят в ветку «dev» и создают свою собственную ветку, в которой будут выполнять определенный ранее функционал проекта;
- при выполнении функционала учащийся делает коммит изменений в файлах с помощью команд «git add», «git commit»;
- после выполнения функционала, учащийся делает «merge» через «Pull Request» из ветки разработки в ветку «dev»;
- при возникновении конфликтов «merge conflicts», учащемуся необходимо разрешить их и продолжить «merge»;
- когда проект полностью выполнен, командир команды делает «merge» из ветки «dev» в ветку «master» и загружает выполненный проект на «gh-pages».

В конце каждого проекта, наступает процесс взаимного оценивания работ – семинарское занятие. Главная цель данного занятия, получение навыков в области тестирования веб-приложений, получение комментариев по возможному улучшению веб-приложения и получение опыта от других участников образовательного процесса.

После выполнения первого проекта «Портфолио» начинается семинарское занятие, в ходе которого учащийся будет оценивать четыре работы своих одноклассников в строгом соответствии с критериями оценивания (Таблица 1):

Таблица 1 – Критерии оценивания проекта «Портфолио»

Критерии/ Оценка	0 баллов	5 баллов	10 баллов	15 баллов
Наличие адаптивной верстки	Верстка не адаптивна	Верстка адаптивна на разрешении 1024px и выше	Верстка адаптивна на разрешении 768px и выше	Верстка адаптивна на разрешении 320px и выше
Применение системы контроля версий GIT	Система контроля версий не задействована	Система контроля версий задействована и имеет менее 15 коммитов	Система контроля версий задействована и имеет 15 – 25 коммитов	Система контроля версий задействована и имеет более 26 коммитов
Кросс-браузерность (Chrome, Opera, Firefox, Safari, Yandex)	Веб-приложение работает по-разному во всех браузерах	Веб-приложение работает в 2 браузерах одинаково	Веб-приложение работает в 3 – 4 браузерах одинаково	Веб-приложение работает во всех браузерах одинаково

Продолжение таблицы 1

Соблюдение семантики	Семантика не соблюдена	Использовано менее трех семантических тега	Использовано три семантических тега	Использовано более трех семантических тегов
Прохождение валидации W3C	В проекте более 3 ошибок	В проекте 2 ошибки	В проекте 1 ошибка	В проекте отсутствуют ошибки
Наличие функционала приложения на смартфонах, компьютерах	Функционал отсутствует	Функционал присутствует на смартфонах	Функционал присутствует на компьютерах	Функционал присутствует на всех платформах

Максимальное количество баллов: 90. Отлично: 90 – 80 баллов, хорошо: 75 – 65 баллов, удовлетворительно: 60 и менее баллов.

Назначение работ для проверки производится случайным образом. Если учащегося проверило четыре человека, то наименьшая оценка отбрасывается и выставляется среднее между тремя оставшимися оценками. В случае, если учащегося проверило менее четырех человек, то при выставлении среднего учитываются все полученные оценки. Количество проверяемых работ может меняться в зависимости от числа учащихся.

После выполнения командного проекта начинается второе семинарское занятие, в ходе которого каждый член команды проверяет проекты других команд в строгом соответствии с критериями оценивания (Таблица 2):

Таблица 2 – Критерии оценивания проекта в команде

Критерии/ Оценка	0 баллов	1 балл	2 балла	3 балла
Наличие адаптивной верстки	Верстка не адаптивна	Верстка адаптивна на разрешении 1024px и выше	Верстка адаптивна на разрешении 768px и выше	Верстка адаптивна на разрешении 320px и выше
Применение системы контроля версий GIT	Система контроля версий не задействована	Система контроля версий задействована и имеет менее 15 коммитов	Система контроля версий задействована и имеет 15 – 25 коммитов	Система контроля версий задействована и имеет более 26 коммитов
Кросс-браузерность (Chrome, Opera, Firefox, Safari, Yandex)	Веб-приложение работает по-разному во всех браузерах	Веб-приложение работает в 2 браузерах одинаково	Веб-приложение работает в 3 – 4 браузерах одинаково	Веб-приложение работает во всех браузерах одинаково

Критерии/ Оценка	0 баллов	1 балл	2 балла	3 балла
Соблюдение семантики	Семантика не соблюдена	Использовано менее трех семантических тега	Использовано три семантических тега	Использовано более трех семантических тегов
Прохождение валидации W3C	В проекте более 2 ошибок	В проекте 2 ошибки	В проекте 1 ошибка	В проекте отсутствуют ошибки
Наличие функционала приложения	Функционал отсутствует	Функционал присутствует на 1 странице	Функционал присутствует на 1-2 страницах	Функционал присутствует на всех страницах

Максимальное количество баллов: 90. Отлично: 90 – 75 баллов, хорошо: 70 – 55 баллов, удовлетворительно: 50 и менее баллов.

Назначение проектных работ для оценки производится случайным образом. Если количество членов команды равняется четырем, то наименьшая оценка за все пункты критериев отбрасывается и выставляется среднее между тремя оставшимися оценками. Если в команде три учащихся, то при выставлении среднего учитываются все оценки. В зависимости от количества команд, если команду проверило больше одной команды, то вычисляется среднее между выставленными оценками.

Ход семинарских занятий разбит на несколько фаз:

- первая фаза – фаза представления работ. На данном этапе учащиеся предоставляют ссылки на выполненные проекты;
- вторая фаза – фаза оценивания. На данном этапе учащиеся оценивают работы в строгом соответствии с критериями оценивания;
- третья фаза – фаза оценивания оценок. На данном этапе учащиеся получают средний балл за выполненный проект и комментарии по возможному улучшению проекта.

Заключение

Успешное прохождение курса формирует практические навыки веб-разработки и дает хороший старт для дальнейшего развития учащегося в качестве веб-разработчика.

Технологии непрерывно развиваются, таким образом разработанный курс также должен поддерживать свою актуальность и постоянно модернизироваться. Для поддержки актуальности, в дальнейших планах проектирование и разработка образовательной платформы, которая будет аккумулировать в себя практико-ориентированные задачи и проекты, расширяющие практический кругозор веб-разработчика. Платформа может создаваться при непосредственном активном практическом участии обучаемых, что позволит сохранять ресурс в актуальном состоянии.

Список литературы

1. Бергер, Е.Г. Современные проблемы и методы прикладной информатики и развития информационного общества: учебное пособие / Е.Г. Бергер. – Москва: РТУ МИРЭА, 2022. – 80 с. – Режим доступа: для авториз. пользователей. – URL: <https://e.lanbook.com/book/239933> (дата обращения: 15.05.2022). – Текст: электронный.

2. Зуфарова, А.С., Суходуб, Р.А. Методика обучения программированию учащихся: проблемы и решения // Управление образованием: теория и практика. 2022. № 4 (50). URL: <https://cyberleninka.ru/article/n/metodika-obucheniya-programmirovaniyu-uchaschihsya-problemy-i-resheniya> (дата обращения: 15.05.2022). – Текст: электронный.

3. Лапчик, М.П., Рагулина, М.И., Семакин, И.Г., Хеннер, Е.К. Методика обучения информатике: учебное пособие / М.П. Лапчик, М.И. Рагулина, И.Г. Семакин, Е.К. Хеннер; под редакцией М.П. Лапчика. – 3-е изд., стер. – Санкт-Петербург: Лань, 2020. – 392 с. – Режим доступа: для авториз. пользователей. – URL: <https://e.lanbook.com/book/139269> (дата обращения: 22.06.2022). – ISBN 978-5-8114-5280-4. – Текст: электронный.

4. Андреев, В.Е., Проблема формирования готовности к профессиональной мобильности у будущих специалистов / В.Е. Андреев // Вестник Череповецкого государственного университета. 2012. № 1 (42). URL: <https://cyberleninka.ru/article/n/problema-formirovaniya-gotovnosti-k-professionalnoy-mobilnosti-u-buduschih-spetsialistov> (дата обращения: 26.06.2022). – Текст: электронный.

5. Голубничий, А.А. Современные методы и средства обучения программированию / А.А. Голубничий, К.А. Чернявская // Бюллетень науки и практики. – 2018. – № 6. – С. 368-372. – Режим доступа: для авториз. пользователей. – URL: <https://e.lanbook.com/journal/issue/308121> (дата обращения: 02.07.2022). – ISSN 2414-2948. – Текст: электронный.

6. Павлов, Л.А. Структуры и алгоритмы обработки данных: учебник для вузов / Л.А. Павлов, Н.В. Первова. – 3-е изд., стер. – Санкт-Петербург: Лань, 2021. – 256 с. – Режим доступа: для авториз. пользователей. – URL: <https://e.lanbook.com/book/156929> (дата обращения: 10.07.2022). – ISBN 978-5-8114-7259-8. – Текст: электронный.

7. Никольский, А.П. JavaScript на примерах: учебное пособие / А.П. Никольский. – Санкт-Петербург: Наука и Техника, 2017. – 272 с. – Режим доступа: для авториз. пользователей. URL: <https://e.lanbook.com/book/101549> (дата обращения: 12.07.2022). – ISBN 978-5-94387-735-3. – Текст: электронный.

8. Самостоятельная работа студентов: методические рекомендации / под редакцией О.В. Дыбиной. – Тольятти: ТГУ, 2011. – 71 с. – Режим доступа: для авториз. пользователей. – URL: <https://e.lanbook.com/book/140302> (дата обращения: 18.07.2022). – Текст: электронный.

Gordeev D.A.

FORMATION OF PRACTICAL SKILLS FOR STUDENTS OF TECHNICAL DIRECTIONS WHEN STUDYING WEB PROGRAMMING

The article proposes an approach to teaching web programming based on the practical application of the studied material, project work in a team and mutual evaluation of project work. The aim of the study is to develop a course on the formation of practical skills for students of technical areas in the study of web programming. The insufficient development of the methodological foundations for the formation of practical competencies in web programming indicates the relevance of the study. The study involved methods of analysis, project activities and mutual evaluation of project activities. As a result of the study, theoretical sections on a number of web technologies have

been systematized, a platform for the algorithmic development of students' skills has been chosen, a web application has been developed that analyzes progress in performing algorithmic tasks, training projects and technical specifications for them have been developed, criteria for evaluating design work have been formulated, and the stages of seminars have been described. peer assessment sessions.

Keywords: *Web-programming, project activities, peer review, algorithms, javascript, git, github, api*