Коллективные проекты по Web-программированию

Колганова В.П. ([Valeriaafanaseva@mail.ru](mailto:Valeriaafanaseva@mail.ru)), Лопатин А.К. ([ak\_lopatin@mail.ru](mailto:ak_lopatin@mail.ru))

ГОУ ВО МО «Государственный социально-гуманитарный университет», г.о. Коломна

Аннотация

Использование метода проектов в системе высшего образования является одним из наиболее естественных способов составления студентом портфолио работ в сфере IT. При должном содержании курса и подогреваемом интересе студентов вполне возможна разработка и поддержка онлайн-приложений, жизненный цикл которых позволит решить сгенерировать естественным способом набор практических задач как для текущего так и будущих потоков студентов. В работе представлен набор предметных областей, а также стек актуальных технологий, освоение которых поможет в развитии и совершенствовании навыков веб-программирования

В образовании последние 5 лет прослеживается несколько трендов: повышение престижа IT-специализаций, повышение взаимодействия бизнеса и образования, рост интереса к искусственному интеллекту и средствам обработки больших данных [3]. Достаточно прозаичен следующий факт: создание студентом сольного хотя бы отчасти конкурентно-способного проекта в области цифровых технологий практически невозможно. Более того, студент, привыкший выполнять одновременно роли дизайнера, программиста, и архитектора с большим трудом будет адаптироваться к работе в команде, роли в которой были распределены за долгое время до его вхождения в неё.

Таким образом при подготовке тем и плана работы над проектом необходимо учитывать ряд вопросов [6].

* разделение frontend и backend (рис. 1);
* определение архитектуры приложения;
* определение ресурсов и площадки для хранения;
* целевая аудитория.

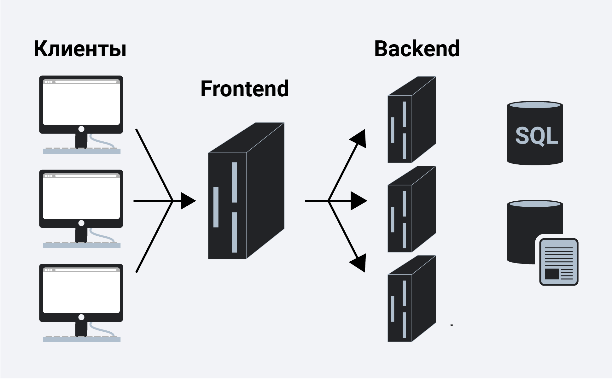


Рис. 1. Клиент-серверная архитектура.

При тщательной подготовке такой ресурс становится цифровым аналогом коллективного творческого дела, ведение которого способствует развитию предметных и метапредметных компетенций [1].

Основой генерации направлений проектов является таких проектов могут являться рядовые операции, проводимые чаще всего участниками образовательного процесса, а также бытовые действия пользователей персонального компьютера:

* редактирование и оформление документов;
* генерация, обработка и анализ изображений;
* обработка рядов числовых данных.

Направления проектов, а именно разнородность входных и выходных данных, а также направленность на Web ограничивают список возможных языков программирования до минимума: это Java, Python или параллельное использование нескольких языков для оптимального решения различных задач. При этом в целом для решения и Frontend и Backend задач может использоваться Python. Отчасти backend может быть описан и на других языках программирования и интегрирован в общее приложение посредством различных интерфейсов [4].

В качестве фреймворков для современной онлайн разработки часто используются следующие: Flask, FastAPI, Django. Отметим, что большая популярность Flask и Django привела к тому, что по этим языкам есть большое количество документации как на русском, так и на иностранных языках. При этом у FastAPI есть свои преимущества, обусловленные именно его реализацией – фреймворк полноценно «из коробки» поддерживает асинхронное программирование, легко справляется с простыми проектами и интеграция отдельных микросервисов(приложений) в систему происходит достаточно просто. Таким образом в случае работы с FastApi студенты-разработчики могут столкнуться с одной стороны – с нехваткой документации, а с другой стороны – с возможностью независимой разработки отдельных сервисов, мгновенной проверкой их работоспособности и легкой интеграцией их в одно большое приложение.

В качестве среды программирования при таком подходе может использоваться любая IDE, однако определенным «стандартом» разработки является владение VS Code или Visual Studio (в том числе из-за обилия плагинов, существенно упрощающих процесс разработки). Также обязательным условием является использование системы контроля версий (SVN): Gitlab, Github или аналог. Ключевое преимущество в том, что ведущих разработчик (его роль может выполнять преподаватель или наиболее «продвинутый» студент) может принять решение о добавлении/изменении определенных строк кода, откате до предыдущей редакции системы и т.п. Более того, сам учащийся может создавать практически бесконечное количество версий своего фрагмента кода и тем самым избавиться от проблемы «удаления нужных файлов» [2].

Отметим, что от преподавателя, решившего использовать такой подход требуется дополнительная подготовка (единовременная): организация репозитория, подготовка структуры проекта (вплоть до планирования структуры каталога на сервере), подготовка и планирование развития проекта (хотя в базовой версии можно каждый год/5 лет сбрасывать сервис к начальному состоянию – это одно из преимуществ SVN), а также наличие знаний используемых технологии, и владение, хотя бы на общем уровне, предметными областями из которых генерируются проекты [7].

Таким образом, используя такой подход, преподаватель веб-программирования получает не только возможность, но и необходимость совершенствовать компетентность в области IT, а образовательная организация – возможность развивать свой сервис/продукт.

Литература

1. Алексеев А.. Введение в Web-дизайн. Учебное пособие // М.: ДМК Пресс, 2019. С. 184.
2. Гарретт Д. Веб-дизайн. Элементы опыта взаимодействия / Д. Гарретт // СПб.: Символ-плюс, 2016. С. 192.
3. Кириченко А. HTML5 + CSS3. Основы современного WEB-дизайна // М.: Просвещение, 2018. С. 104.
4. Кирсанов Д. Веб-дизайн: книга Дмитрия Кирсанова / Д. Кирсанов // М.: Символ, 2015. С. 368.
5. Макнейл П. Веб-дизайн. Книга идей веб-разработчика / П. Макнейл // СПб.: Питер, 2017. С. 480.
6. Пилгрим М. Погружение в HTML5 // М.: Просвещение, 2019. С. 271.