2.1 Среда разработки – WebStorm

**WebStorm** представляет собой инструмент для разработки web-сайтов и редактирования HTML, CSS и JavaScript кода. Решение обеспечивает быструю навигацию по файлам и генерирует уведомления о возникающих проблемах в коде в режиме реального времени. JetBrains WebStorm позволяет добавлять разметку HTML-документов или элементов SQL непосредственно в JavaScript. JetBrains WebStorm осуществляет развертывание и синхронизацию проектов через протокол FTP.  
  
Используя возможности кода HTML/XHTML и XML, WebStorm обеспечивает автоматическое завершение стилей, ссылок, атрибутов и других элементов кода. При работе с CSS осуществляется завершение кода классов, HTML-номеров, ключевых слов и т. д. WebStorm предлагает автоматическое решение таких проблем, как выбор формата, свойств, классов, ссылок на файлы и других атрибутов CSS. Решение позволяет использовать мощность инструмента Zen coding для верстки HTML, отображает действия тэга на web-странице. Продукт WebStorm осуществляет завершение кода JavaScript для ключевых слов, лейблов, переменных, параметров и функций DOM и поддерживает специфические особенности популярных браузеров. Реализованные в решении функции рефакторинга JavaScript позволяют преобразовывать структуру кода и файлов и .js.

Основные фреймворки:

Vue.js — это JavaScript библиотека для создания веб-интерфейсов с использованием шаблона архитектуры [MVVM](https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel) (Model-View-ViewModel).

Функции Vue.js:

* Реактивные интерфейсы;
* Декларативный рендеринг;
* Связывание данных;
* Директивы (все директивы имеют префикс «V-». В директиву передается значение состояния, а в качестве аргументов используются html атрибуты или Vue JS события);
* Логика шаблонов;
* Компоненты;
* Обработка событий;
* Свойства;
* Переходы и анимация CSS;
* Фильтры.

2. Node.js — это платформа, основанная на [среде выполнения Chrome JavaScript,](https://translate.googleusercontent.com/translate_c?depth=1&rurl=translate.google.ru&sl=en&sp=nmt4&tl=ru&u=https://code.google.com/p/v8/&xid=17259,15700022,15700186,15700190,15700256,15700259,15700262&usg=ALkJrhgKsMZ3b8T_FL5p_HVQ6X6IpJvxKw) для простого создания быстрых и масштабируемых сетевых приложений. Node.js использует управляемую событиями неблокирующую модель ввода / вывода, которая делает ее легкой и эффективной, идеально подходящей для приложений с интенсивным использованием данных в реальном времени, работающих на распределенных устройствах.

3. ExpressJS — это структура веб-приложений, которая предоставляет вам простой API для создания веб-сайтов, веб-приложений и серверных частей. С ExpressJS вам не нужно беспокоиться о протоколах низкого уровня, процессах и т. Д.

Express предоставляет минимальный интерфейс для создания наших приложений. Он предоставляет нам инструменты, необходимые для создания нашего приложения. Это гибкий инструмент , так как на npm доступно множество модулей, которые можно напрямую подключить к Express.

2.3 Программные классы – User, Course, Tests, Question, Answer

Ниже представлен код работы программных классов:

title: '',

description: '',

video: '

async addCourse () {

await PostsService.saveCourse({

title: this.title,

description: this.description,

video: this.video

})

this.$swal(

'Great!',

'Your post has been added!',

'success'

)

await this.$router.push({name: 'Posts'})

}

async Viewing () {

const response = await PostsService.getData()

this.posts = response.data.posts

},

},

saveCourse (params) {

return Api().post('add\_post', params)

},

2.4 Функции с учётом особенностей взаимодействия с СУБД:

Ниже представлен код работы функций с учётом особенностей взаимодействия с СУБД:

exports.create = (req, res) => {

var db = req.db;

var title = req.body.title;

var description = req.body.description;

var video = req.body.video;

var new\_post = new Post({

title: title,

description: description,

video: video

})

new\_post.save(function (error) {

if(error) {

console.log(error)

}

res.send({

success: true

})

})

};

exports.findAll = (req, res) => {

Post.find({}, 'title description video', function (error, posts) {

if (error) {

console.error(error);

}

res.send({

posts: posts

})

}).sort({\_id: -1})

};

exports.findOne = (req, res) => {

var db = req.db;

Post.findById(req.params.id, 'title description video', function (error, post) {

if (error) {

console.error(error);

}

res.send(post)

})

};

exports.delete = (req, res) => {

var db = req.db;

Post.remove({

\_id: req.params.id

}, function (err, post) {

if (err) {

res.send(err)

}

res.send({

success: true

})

})

};

exports.UpdateUser = (req, res) => {

var db = req.db;

Post.findById(req.params.id, 'title description video', function (error, post) {

if (error) {

console.error(error);

}

post.title = req.body.title

post.description = req.body.description

post.video = req.body.video

post.save(function (error) {

if (error) {

console.log(error)

}

res.send({

success: true

})

})

})

};

Библиотеки:  
Библиотека [Axios](https://github.com/axios/axios), предназначенная для выполнения HTTP-запросов, основана на промисах. Она подходит для использования в среде Node.js и в браузерных приложениях. Библиотека поддерживает все современные браузеры

import axios from 'axios'

Router — официальная библиотека маршрутизации для [Vue.js](https://ru.vuejs.org/). Она глубоко интегрируется с Vue.js и позволяет легко создавать SPA-приложения. Включает следующие возможности:

* Вложенные маршруты/представления
* Модульная конфигурация маршрутизатора
* Доступ к параметрам маршрута, query, wildcards
* Анимация переходов представлений на основе Vue.js
* Удобный контроль навигации
* Автоматическое проставление активного CSS класса для ссылок
* Режимы работы HTML5 history или хэш, с авто-переключением в IE9
* Настраиваемое поведение прокрутки страницы

import Router from 'vue-router’

3) VueSweetalert2 позволяет разработчикам создавать пользовательские всплывающие окна для показа своим пользователям простой установкой значений нескольких свойств. Это руководство предназначалось для изучения основ этой библиотеки, чтобы вы могли быстро создавать свои собственные оповещения.

import VueSweetalert2 from 'vue-sweetalert2'

4) Mongoose - это библиотека JavaScript, часто используемая в приложении Node.js с базой данных MongoDB. В данной статье я собираюсь познакомить вас с Mongoose и MongoDB и, что более важно, показать, где их уместно использовать в приложении.

const mongoose = require('mongoose');

5) Express, фреймворк web-приложений для Node.js, реализованный как свободное и открытое программное обеспечение под лицензией MIT. Он спроектирован для создания веб-приложений и API

var express = require("express");

6) body-parser анализирует кодированные данные JSON, buffer, string и URL, представленные с помощью запроса HTTP POST

var bodyParser = require("body-parser");

7) CORS (Cross-Origin Resource Sharing) механизм, который использует дополнительные заголовки HTTP, чтобы дать браузерам указание предоставить веб-приложению, работающему в одном источнике, доступ к ответу на запрос к ресурсам из другого источника.

var cors = require('cors');