**Other Frameworks**

Research has been conducted on multiple web application language types. Each language has a small description giving a basic idea of how it works, or what it is like, what it is used for, it’s pros and cons, and examples of where it has been used in industry. The resources I used to create these notes are also featured in each language table.

**Front End**

|  |  |
| --- | --- |
| **Vue.js** | |
| **Description** | **JavaScript** framework for building user interfaces  **Declarative** and **component-based** programming model  Builds on top of **HTML, CSS and JavaScript** |
| **Uses** | **Front end**  Creating **interactive web interfaces** and **single page applications**  Handles **data updates** and **navigation** **without reloading** the page |
| **Pros** | **Easy to Learn**  **High Performance**   * *Uses virtual* ***DOM*** *(Document Object Model)*   **Extensive Documentation**  **Fast Reaction**   * *Useful for* ***real-time update*** *requirements*   **Flexibility** |
| **Cons** | **Difficulties with two-way binding**  **Lack of plugins**  **Less useful tutorials in English** |
| **Examples** | *Facebook, Netflix, Trivago, GitLab* |
| **Resources** | [Vue.js website](https://vuejs.org/), [Pros and Cons](https://ddi-dev.com/blog/programming/the-good-and-the-bad-of-vue-js-framework-programming/) |

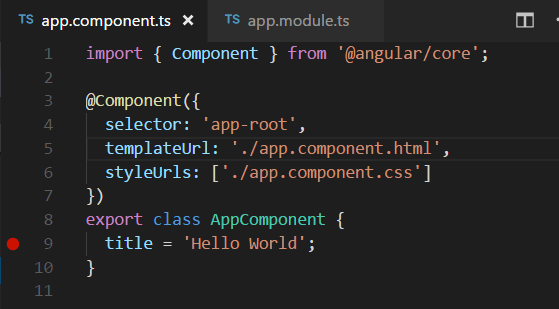
**Code example:**

A screenshot of a computer code

Description automatically generated

|  |  |
| --- | --- |
| **Angular** | |
| **Description** | **JavaScript** Framework written in **TypeScript**   * *Allows the addition of* ***types*** *to data*   **MVC framework** |
| **Uses** | **Front end**  **Single page** applications |
| **Pros** | **Uses Jasmin for testing**  **Well-integrated libraries** |
| **Cons** | **Steep Learning curve**  **Complex** |
| **Examples** | *Mircosoft Office, Gmail, Samsung* |
| **Resources** | [Angular Website](https://angular.io/guide/what-is-angular), [Pros and Cons](https://www.simplilearn.com/tutorials/angular-tutorial/what-is-angular) |

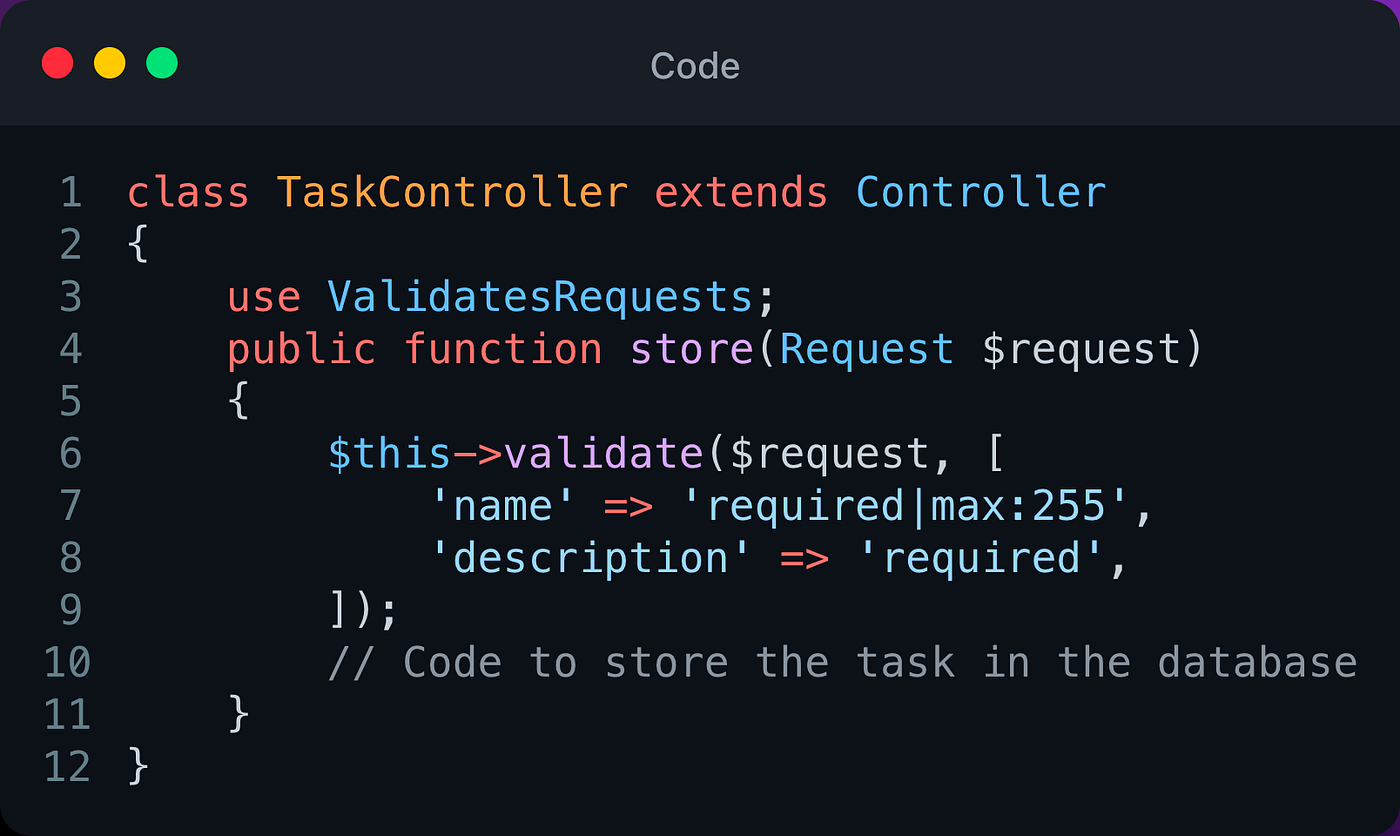
**Code example:**



**Back End:**

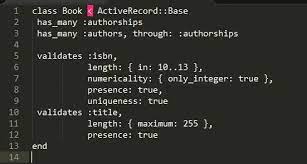
|  |  |
| --- | --- |
| **Laravel** | |
| **Description** | **Uses PHP**  **MVC Structure** |
| **Uses** | **Back End**  Can be used with **Vue/React** through Inertia  Good for **booking web apps** and **Customer Relationship Management** systems |
| **Pros** | **Simple User Authentication**  **Scalable**  **Extensive Documentation,** *including tutorials and videos*  **Laravel Dusk is available to help with automated browser tests** |
| **Cons** | **Slower development compared to other frameworks** |
| **Examples** | *ALISON (e-learning)* |
| **Resources** | [Laravel Website](https://laravel.com/), [Pros and Uses](https://linkupst.com/blog/laravel-pros-and-cons-for-web-app), [More Uses](https://linkupst.com/blog/laravel-pros-and-cons-for-web-app) |

**Code Example:**



|  |  |
| --- | --- |
| **Ruby On Rails (RoR)** | |
| **Description** | Open Source Framework for Web Development in **Ruby**  **OOP language** like Python  **MVC** (*Model, view, controller*) framework, all **layers** are provided by **Rails** (*see* [*RoR Guide*](https://guides.rubyonrails.org/) *for examples and explanations*)  Can also use MySQL |
| **Uses** | **Back end**  Rendering **HTML templates** and maintain **live pages** via **WebSockets**  Updating **databases**  Sending and receiving **emails** |
| **Pros** | **Simple and easy to learn**   * *If our application is* ***expanded upon****, new developers can be* ***trained quickly****.*   **Everything built into one due to the MVC framework** |
| **Cons** | **Inflexible**  **Slower performance**   * *We would have to keep applications reasonably* ***simple*** *to conform to RoR’s needs* |
| **Examples** | *GitHub, Airbnb, Kickstarter, Instacart (food shopping)* |
| **Resources** | [RubyOnRails Website](https://rubyonrails.org/), [Pros and Cons](https://www.computer.org/publications/tech-news/build-your-career/developers-insight-into-the-ruby-on-rails), [RoR Guide](https://guides.rubyonrails.org/) |

**Code Example:**



**Arguably front and back end:**

|  |  |
| --- | --- |
| **Node.js** | |
| **Description** | **Asynchronous event** driven  **JavaScript** |
| **Uses** | **Front end and Back end**  **Scalable** network applications |
| **Pros** | **High-Performance real-time applications**  **Scalable**  **Extensibility**  **Built-in APIs for *HTTP, TCP, DNS,…***  **Authentication** |
| **Cons** | **Does not use threads**  **Reduces performance for complex computing**  **Lack of Library Support** |
| **Examples** | *Paypal, Uber, LinkedIn* |
| **Resources** | [NodeJs Website](https://nodejs.org/en), [Pros and Cons](https://www.sayonetech.com/blog/nodejs-advantages-disadvantages/) |

**Code Example:**

