

Technologies

Front-end considerations

HTML, CSS, JS

Basic foundations

WEB APPLICATION FRAMEWORK

Node.js and Express

FRONTEND FRAMEWORK CONSIDERATIONS

React JS library

More efficient: employs a virtual DOM to optimize performance

Standard and familiar (relatively small learning curve)

Modular/reusable elements: component-based architecture

Easier debugging

Improved load time and performance with efficiently built UI

Easy integration with 3rd party libraries

Flexible structures

One-way data binding

Needs third party applications to manage data-binding

More ideal for dynamic, simple web apps

More popular (better support?)

Angular

Has built in data binding (doesn't need to use external libraries)

More efficient with fetching data

FRONTEND LIBRARIES/TOOLS

Redux

Bootstrap

APIs

Design RESTful apis for communication (requesting/sending data) between frontend and backend

Possible external APIs for social media integration

AUTHENTICATION

OAuth

WEB SECURITY?

SSL/TLS

CSRF Protections, Input Validation, Sanitization

TESTING

Unit testing: Jest

End-to-end testing: Cypress/Selenium

VERSION CONTROL

Git

HOSTING/PRESENTING

GitHub Pages

(or GoDaddy if needed)