SOFTWARE ENGINEERING DEFINED

Software engineering is the process of using a programming language (and method) to design a program that runs on a computer to perform or automate a given task.
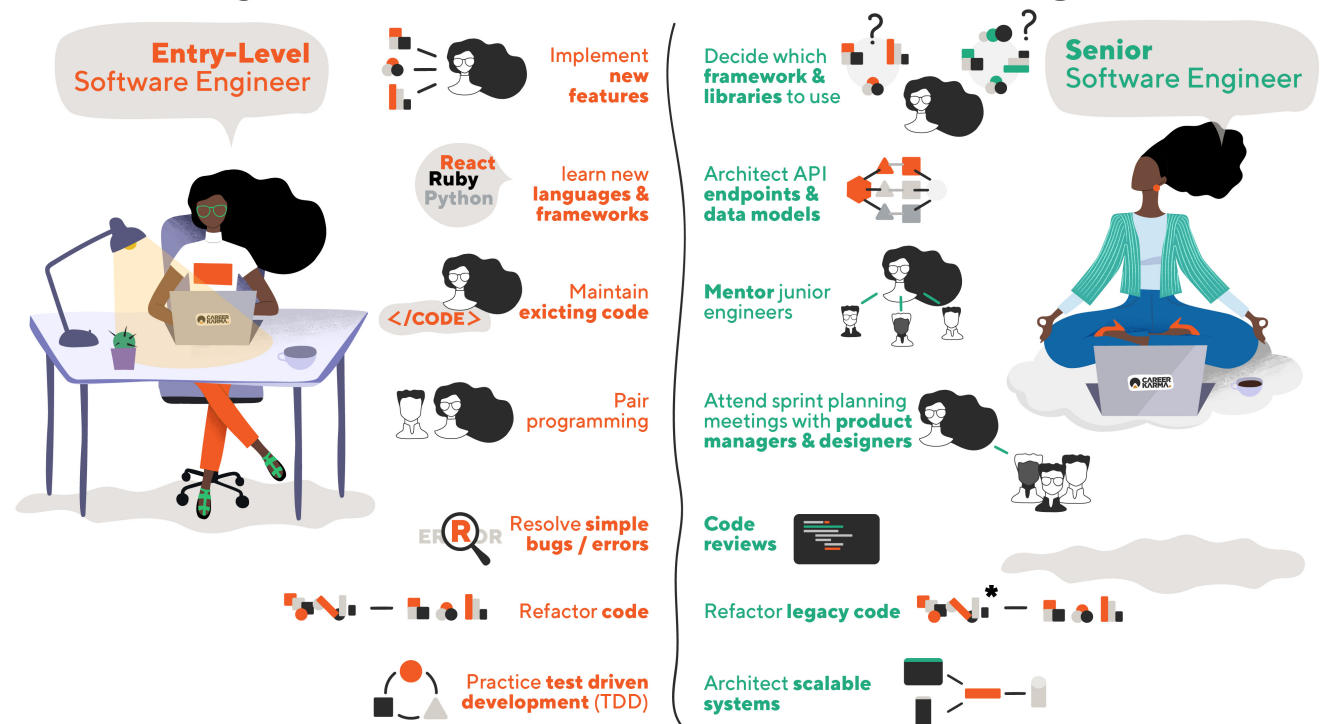
MOST COMMON ROLES & ASSOCIATED BILL RATES

FRONT END DEVELOPER
- React
- Angular
- Vue
- JavaScript (VanillaJS)

FULL STACK DEVELOPER
- React/.NET
- Angular/Java
- React/Node
- Vue/Go

BACK END DEVELOPER
- .NET
- Go
- NodeJS
- Python
- Java

MOBILE DEVELOPER
- iOS
- Android
- React Native

AVERAGE SALARY BANDS

Lead/Architect: $150k+

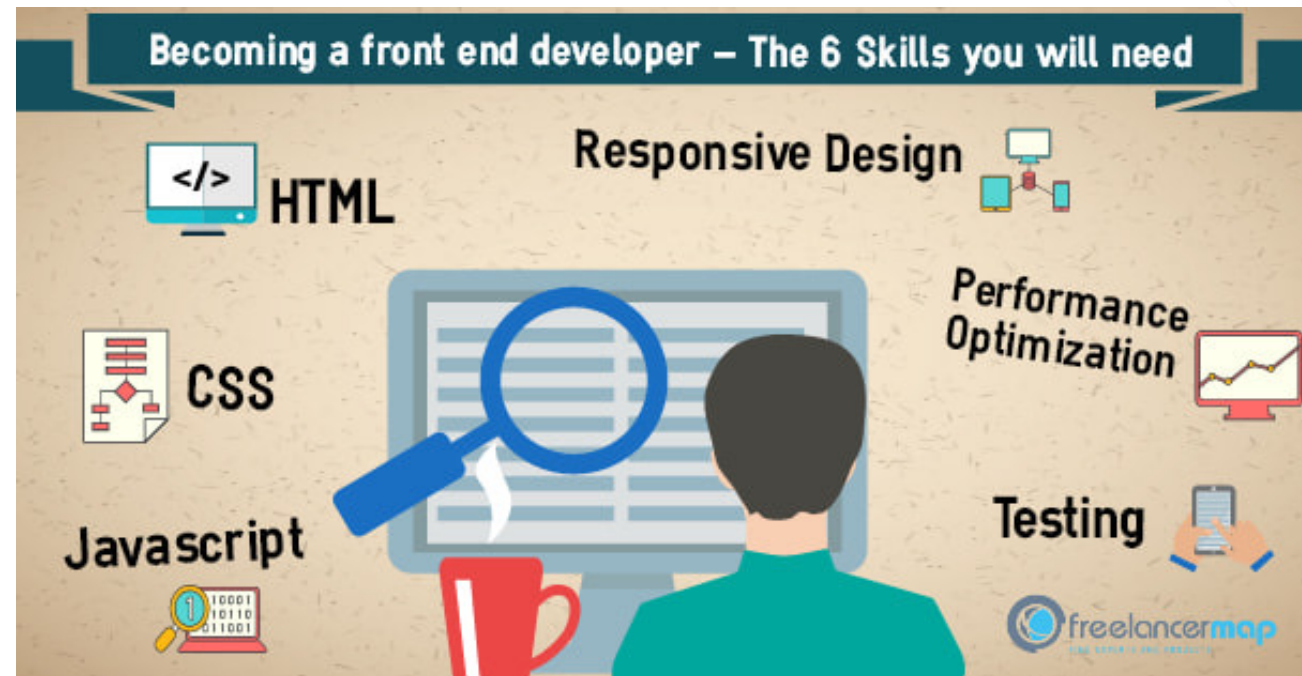Senior: $130k-$140k

Mids: $100k-$120k

Junior-Mid: $80k-$95k

Entry-Junior: $60k-$75k

FRONT END DEVELOPMENT EXPLAINED

Front-end is the part of the code that is on the front of the application. It is usually visible to the user in the form of an interface inviting the user to interact. The main purpose of the front-end code is to interact with the user, as well as present the data in a well-defined style and matter.

- JavaScript
  - React
  - Angular/AngularJS
  - Vue
  - jQuery
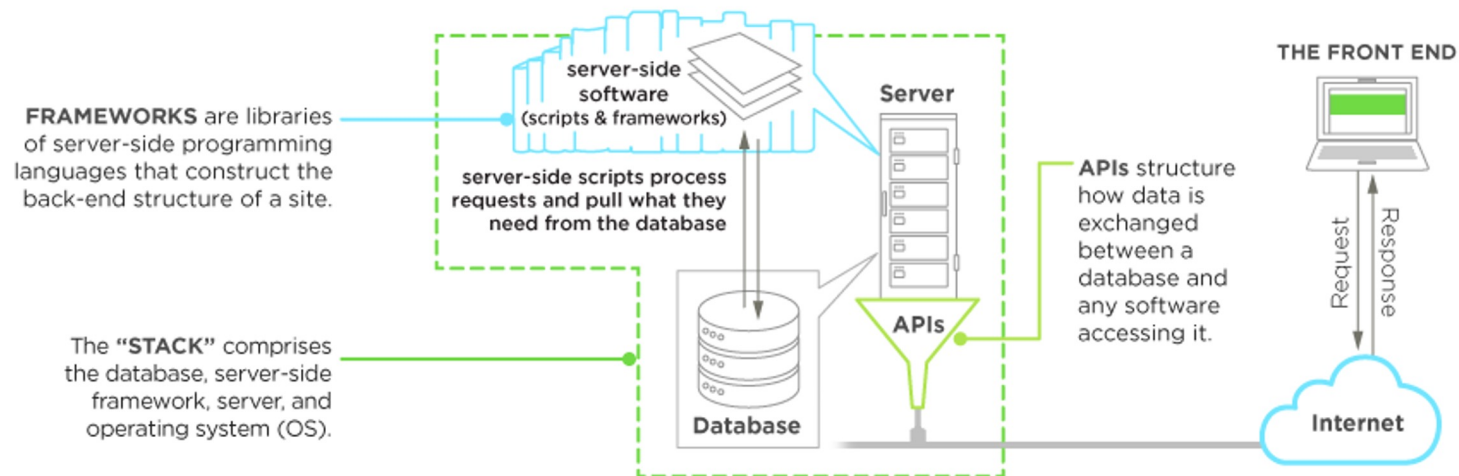- ASP.NET MVC/Core
- HTML/CSS

Becoming a front end developer – The 6 Skills you will need

Responsive Design

HTML

CSS

Performance Optimization

Javascript

Testing

freelancermap

BACK END DEVELOPMENT EXPLAINED

Back-end is the part of the application that is never visible to the user. It is built with use of server-side languages and databases. In simpler words, front-end code interacts with a user in real time, while back-end code interacts with a server to return user ready results. Anything displayed on the website is because of the query performed on the back end returning data to the front-end.

- .NET
- Go
- Python
- NodeJS
- PHP
- Java

BACK-END DEVELOPMENT & FRAMEWORKS IN SERVER SIDE SOFTWARE

FRAMEWORKS are libraries of server-side programming languages that construct the back-end structure of a site.

server-side software (scripts & frameworks)

server-side scripts process requests and pull what they need from the database

Server

APIs structure how data is exchanged between a database and any software accessing it.

The **"STACK"** comprises the database, server-side framework, server, and operating system (OS).

Database

APIs

THE FRONT END

Request / Response

Internet

upwork™

FULL STACK DEVELOPMENT EXPLAINED

Full-stack developers are comfortable working with both back-end and front-end technologies. A full-stack developer doesn't need to master all the areas and technologies needed to work it. The developer just needs to be comfortable working with the technologies associated while delivering at the level expected by the client.
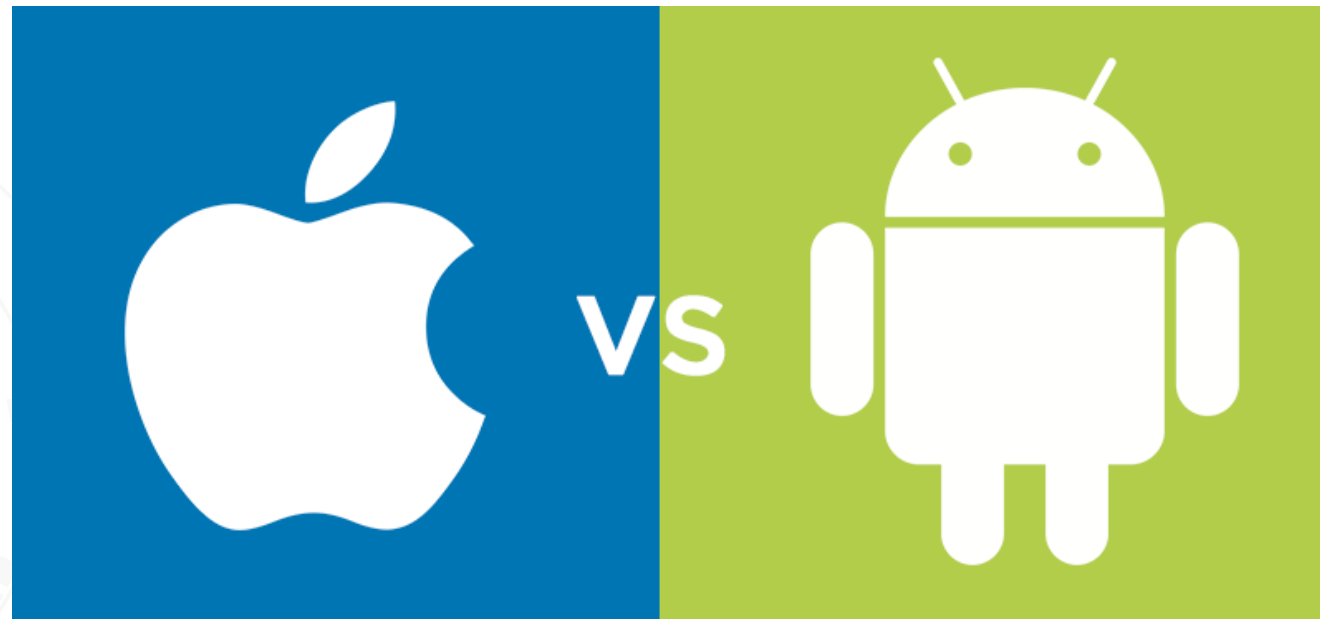
- .NET
- Go
- Python
- NodeJS
- PHP
- Java
- JavaScript
  - React
  - Angular/AngularJS
  - Vue
  - jQuery
- ASP.NET MVC/Core

MOBILE DEVELOPMENT EXPLAINED

Mobile development is software for smartphones and digital assistants, most commonly for Android and iOS. The software can be preinstalled on the device, downloaded from a mobile app store, or accessed through a mobile web browser. The programming languages used for this kind of software development include Kotlin/Java (Android), Swift/Objective-C (iOS), and React Native (JavaScript).

- iOS
- Android
- React Native

```
                          ┌─────────────────┐
                          │  Full Stack Dev │
                          └─────────────────┘
                                   │
        ┌──────────────┬───────────┼───────────┬──────────────┐
        ▼              ▼           ▼           ▼              ▼
┌──────────────┐ ┌──────────────┐ ┌──────────┐ ┌────────┐ ┌──────────────┐
│ Front-end Dev│ │ Back-end Dev │ │Databases │ │ DevOps │ │Mobile App Dev│
└──────────────┘ └──────────────┘ └──────────┘ └────────┘ └──────────────┘
        │              │           │           │              │
        ▼              ▼           ▼           ▼              ▼
   ┌────────┐     ┌─────────┐  ┌──────────┐ ┌────────┐  ┌──────────────┐
   │  CSS   │     │ Python  │  │  MySQL   │ │   CI   │  │ Hybrid Apps  │
   │  JS    │     │ Node JS │  │  SQLite  │ │   CD   │  │ IOS/ Android │
   │  SPA   │     │  PHP    │  │ PostGres │ │  AWS   │  └──────────────┘
   └────────┘     │  GO     │  │ Mongo DB │ └────────┘
      │ │         └─────────┘  └──────────┘     │              │
```

Front-end Dev → CSS / JS / SPA
- LESS, SAAS, GULP
- Angular 2, React, TypeScript

Back-end Dev → Python / Node JS / PHP / GO
- Django, Express JS
- Design, Caching, Middleware

Databases → MySQL / SQLite / PostGres / Mongo DB
- SQL, Clusters, Joins, Sharding

DevOps → CI / CD / AWS
- Nginx, Ansible, Lambda, SQS

Mobile App Dev → Hybrid Apps / IOS/ Android
- React Native, Ionic

HOW TO SPOT A GOOD SOFTWARE ENGINEER

RESUMES

- Good tenure

- Succinct and thorough skills summary
  - Too many technologies listed
  - Random bolded words
  - Terrible formatting

PRESCREENS

- Working on interesting software with many users

- Greenfield development vs. maintenance and support

- Can describe in detail what they work on

- Can hold a conversation, but not talk your ear off

- Passionate – side projects they can speak on

- Ask about their story

GOOD RESUME

Technologies and Languages

- Languages: Go, Bash, Java
- Technologies: Swagger, Git, REST, JSON

Experience

Software Engineer   **IBM**   **Oct 2019 - Present**
IBM Blockchain   Durham, NC

- Active contributor to the Hyperledger Fabric Blockchain DLT open source project on Github, RocketChat, and Stack Overflow
- Designing and building configuration management go library to reduce dependencies required during configuration update process
- Designed and Implemented process to generate Swagger definition
- Implemented channel participation API to improve usability and maintainability around channel management
- Delivered multiple tutorials in Go and JavaScript to clearly demonstrate private data concepts

Software Engineer   **IBM**   **June 2017 - 2019**
Cloud Foundry Enterprise Environment   Durham, NC

- Built distributed micro services for managing and provisioning Cloud Foundry instances on Kubernetes
- Developed and documented SOC2 tooling
- Responsible for planning, development, and delivery of multiple major features
- Developed and managed CI/CD concourse pipeline
- Designed and developed Kibana dashboards to track application statistics
- Security focal ensuring any identified PSIRTs were addressed and resolved

Site Reliability Engineer   **IBM**   **June 2016 - 2017**
IBM Cloud Public and Dedicated   Durham, NC

- Designed and built tools to manage a rapidly growing number of servers and services
- Monitored and triaged issues to ensure the availability and reliability of cloud offerings by keeping critical systems operational
- Responsible for deployment of changes and management of multiple IAAS/PAAS offerings

✔ Key technologies
✔ Concise
✔ Good tenure and path
✔ Reputable company
✔ Consistent city/state*

WAT QUESTIONS TO ASK

FRONT END DEVELOPMENT

- How do you go about scaling your apps for performance optimization?

- What are you using for state management?

- Are you involved in unit testing, or does QA handle most?

BACK-END DEVELOPMENT

- Are you developing any microservices? How many?

- How many users does your application support?

- How involved are you in the CI/CD process?

WAT QUESTIONS TO ASK

FULL STACK DEVELOPMENT

- Are you more back end or front end focused?

- What would you say your split of front vs back end is?

- Which side do you like best?

MOBILE DEVELOPMENT

- What platform are you most comfortable developing in?

- What languages do you work best with?

- Do you have any apps on the App Store or Google Play?

AREAS OF SUCCESS

- NCR
  - Jay Fry – React Developer
  - Brian D'Ostilio – Go Developer
  - Natalie Weinert – C# Developer
  - Tiffany Harris – Go Developer
- rewardStyle
- Obie
- AT&T
- Domino's

FURTHER LEARNING

- [Microservices explained](#)

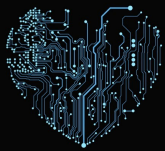- [Full Stack Development](#)

- [Front End vs. Back End](#)