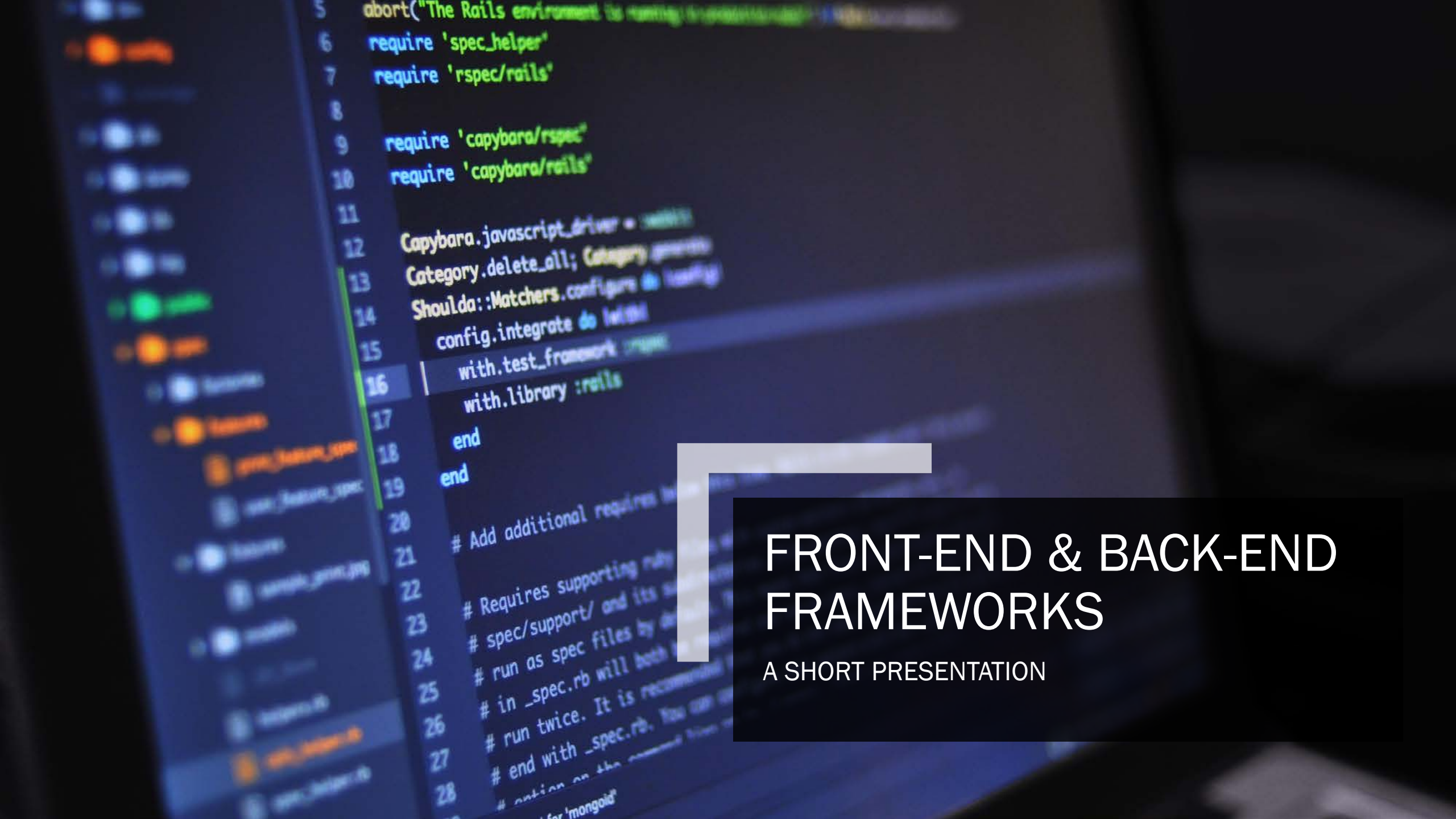

FRONT-END & BACK-END FRAMEWORKS

A SHORT PRESENTATION

Front-End & Back-End

Front-End Development

- Also known as ‘client-side’
- Development of user interactive part of web app
- Tables, Graphs, Buttons, Colors, Nav Menus
- HTML, CSS, JavaScript

Back-End Development

- Also known as ‘server-side’
- Development of invisible part of web app
- Indirectly accessed by users through the Front-End
- Database querying, writing API
- PHP, Python, C++

Full-Stack Developer

- A developer who can develop both client and server side

Frameworks

- A platform that provides a foundation for developing software or web applications.
- A “template” of a working program that is modified to the developer’s need
- Contains common code with generic functionality that can be selectively specialized or overridden

Front-End Frameworks

- React JS
- Angular
- Vue JS
- Bootstrap

Back-End Frameworks

- Django
- Laravel
- Ruby
- Express JS

React JS

- Open-source framework developed by Facebook, for ***JavaScript***

Pros

- Usability
- Huge community support
  as per Stack Overflow

Cons

- Poor documentation
- JSX complexity

Usage Context

- We can use React to build interactive visual layers of single-page web apps and Progressive Web Apps (PWAs) within a short time

Famous Apps

- PayPal, Netflix, Walmart

Angular

- A simple and straightforward framework for *TypeScript* where it has two-way data binding to give components a way of sharing data.

Pros

- Flexibility
- Google support
- High performance
- Extensive size of community

Cons

- Hard to learn
- Heavy due to complexity

Usage Context

- We use it for complex and large scale applications like enterprise based and active web apps

Famous Apps

- Xbox, Blender, BMW

Vue JS

- A progressive framework for ***JavaScript*** to build web interfaces and one-page applications. Also used in desktop and mobile app developments.

Pros

- Detailed documentation
- Easy to learn
- High performance
- Third-party integrations

Cons

- Relatively small community
- Linguistic challenges

Usage Context

- We can use Vue JS to build both small and large-scale web and mobile apps, PWAs and complex single-page apps

Famous Apps

- Alibaba, Reuters, Xiaomi

Bootstrap

- Is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites using ***CSS*** language

Pros

- Supports most browsers
- Flexible and easy to work with
- Open-source and free to use
- Very time saving

Cons

- Non-compliant HTML
- If no customization, all the websites will look the same

Usage Context

- We can use Bootstrap when we want to customize a web page but don’t want to write the CSS from scratch

Famous Apps

- LinkedIn, Udemy, Duolingo

Django

- An open-source and free web framework that is *Python* based
- It follows MTV (model-template-views) architectural pattern

Pros

- Saves time
- Easy to learn
- Huge community
- Secure

Cons

- Not for smaller projects
- Monolithic framework
- Multiple request issue

Usage Context

- We use Django for creation of database-driven, complex websites

Famous Apps

- Mozilla, YouTube, Instagram

Laravel

- An open-source *PHP* framework, which is robust and easy to understand. It follows the MVC design pattern.

Pros

- Very secure
- Plentiful documentation
- Allows individual unit testing
- Data migration is simple

Cons

- Support not readily available
- Frequent update/evolvement
- Easy to learn but quite difficult to master

Usage Context

- We use Laravel when your development team has in-depth knowledge on PHP and you need to manage dynamic content, databases, session tracking or build e-commerce sites

Famous Apps

- BBC, Pfizer

Ruby on Rails

- Written for ***Ruby*** programming language used for developing database-backed web applications. It uses MVC (model-view-controller) design pattern.

Pros

- Built-in security measures
- High productivity
- Huge and active community

Cons

- Not preferred for huge applications
- Not flexible
- Continuous evolvement/version update

Usage Context

- Use RoR if your dev. Team has in-depth knowledge on Ruby
- To create an application quickly with variety of complex features and don’t have a defined concept

Famous Apps

- GitHub, Airbnb, Twitch

Express JS

- Free and open-source framework for ***Node.js*** used in designing web applications and APIs.

Pros

- Provide package for building APIs
- Flexible and simple
- Easy to connect with databases
- Huge & supportive community
- Fast application development

Cons

- Callback issues
- Not big on security
- Error messages usually not helpful

Usage Context

- When we want to build APIs
- When you want simple routing for requests made by clients

Famous Apps

- PayPal, Stack, Twitter

Full-Stack Principles

- Have solid programming language skills
- Have good skills with framework and third-party libraries
- Have basic design ability
- Should be global thinking, good communication skills
- Creative
- Good time management skills