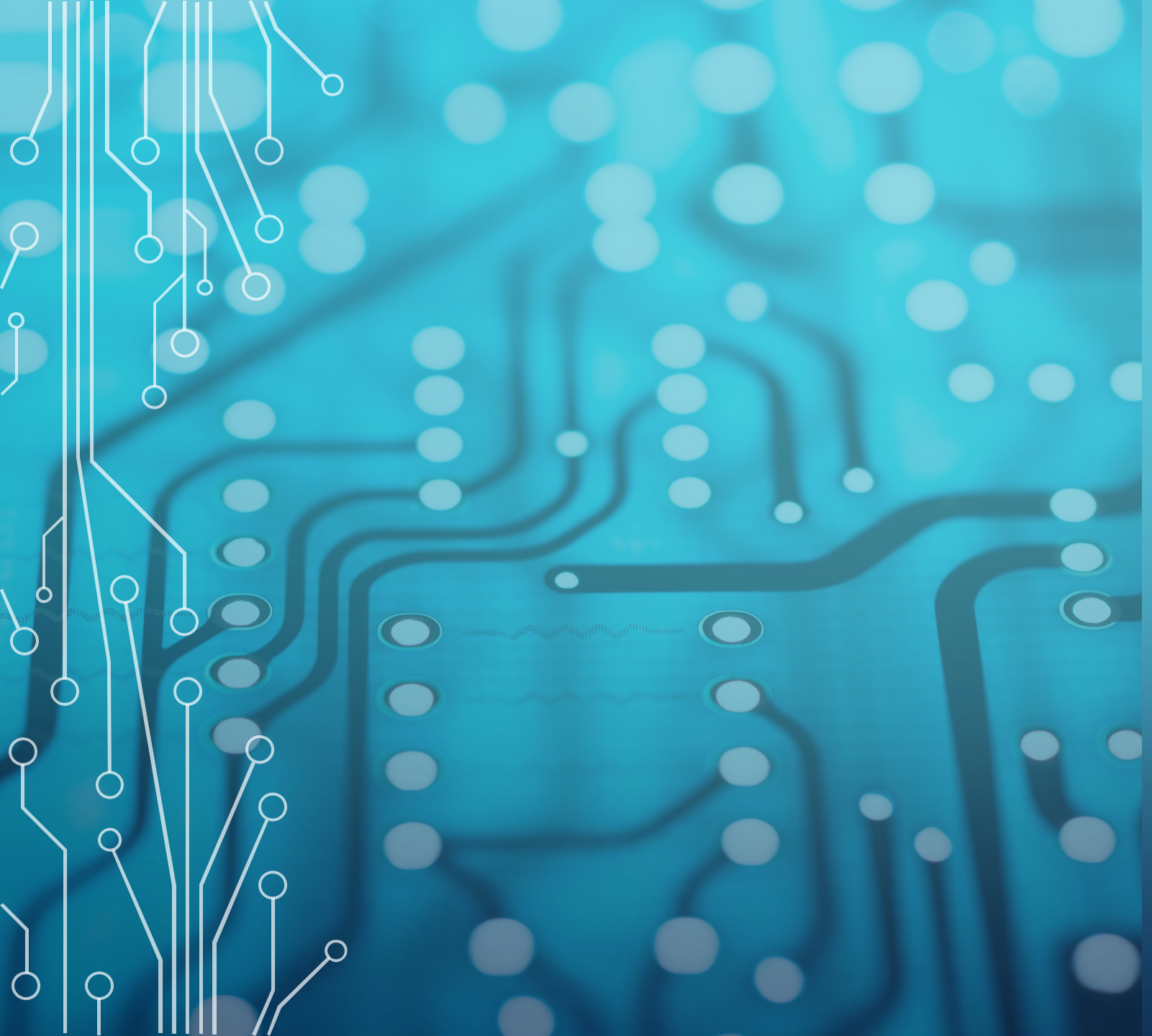
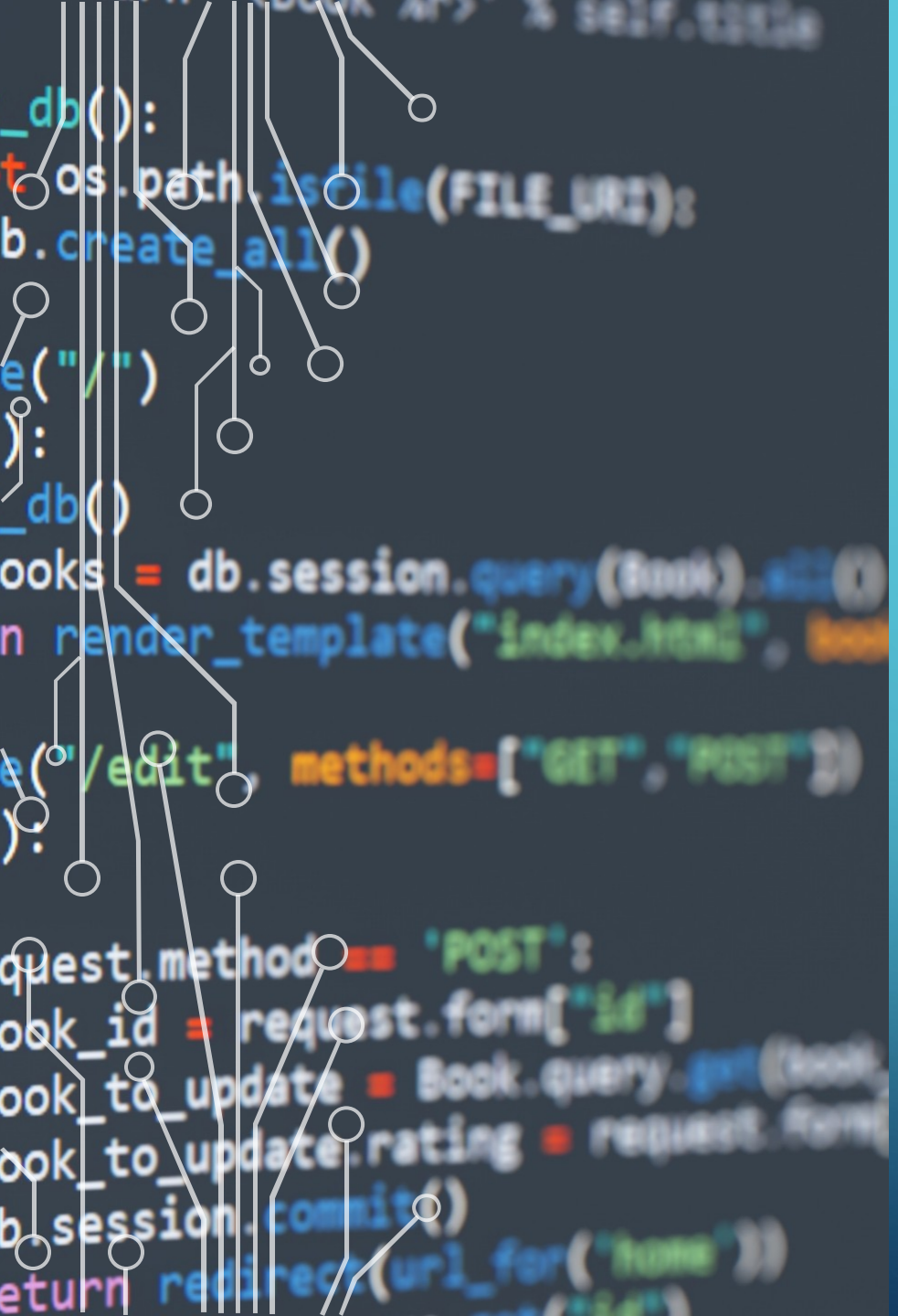
MVC

MODEL-VIEW-CONTROLLER



WHAT WE'RE COVERING

- What is MVC
- Why we use it.
- The components
- ASP.NET MVC Framework



INTRODUCTION TO MVC

- MVC stands for Model-View-Controller, a software architectural pattern.
- It's widely used to structure code in a way that separates concerns and improves maintainability, flexibility and scalability.
- MVC divides an application into three interconnected components, each with a specific responsibility.

COMPONENTS OF MVC - MODEL

- Represents the data and business logic of the application.
- Responsible for data storage, retrieval, and manipulation.
- Independent of user interface and input.

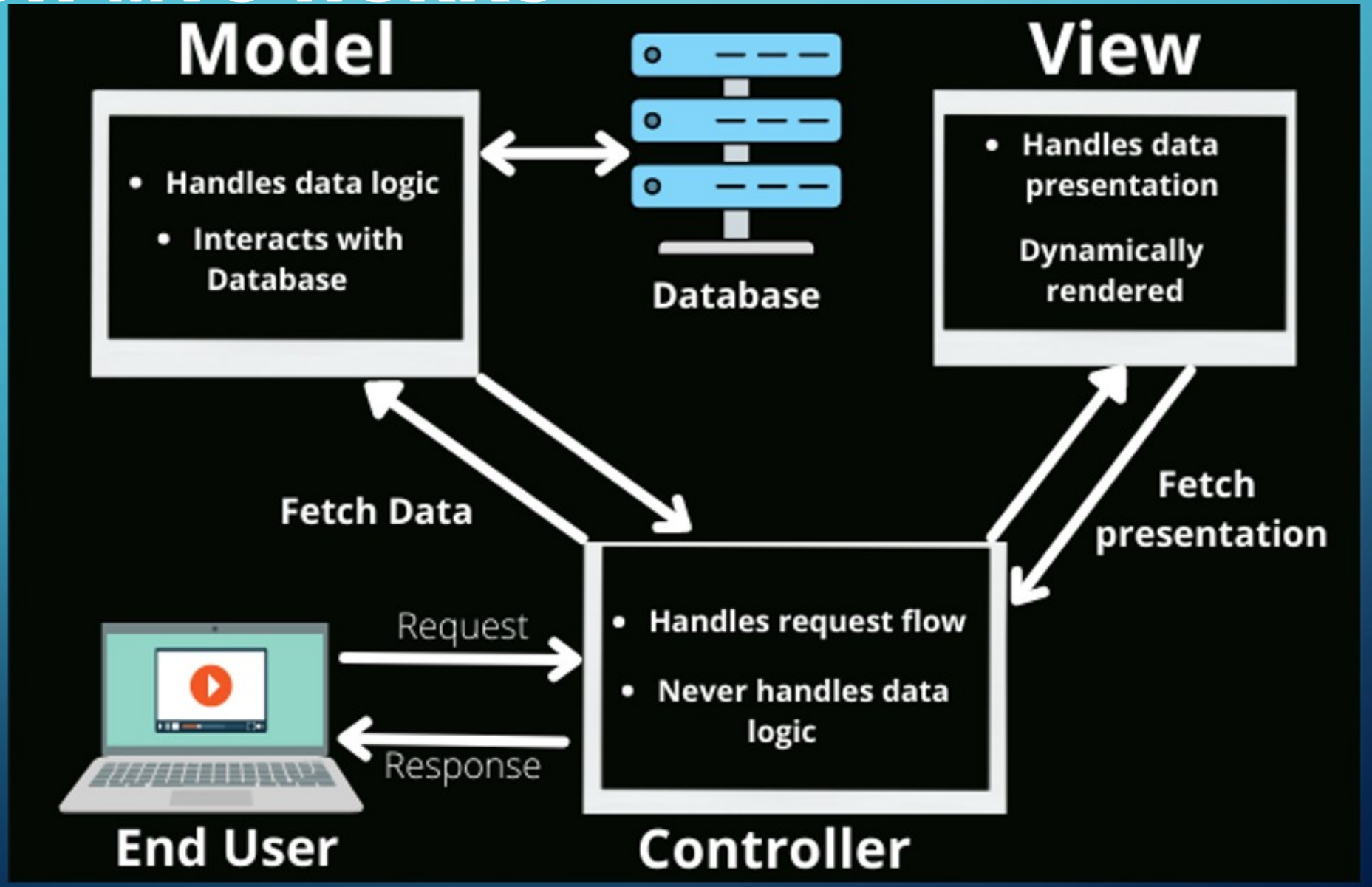
COMPONENTS OF MVC - VIEW

- Represents the user interface and presentation of data.
- Displays information to users.
- Provides forms and input fields for the user to interact with the system.

COMPONENTS OF MVC - CONTROLLER

- Acts as an intermediary between the model and the view.
- Receives and handles all user input, processes it, and updates the model and view accordingly.
- Coordinates the flow of data between the model and the view sections.

HOW MVC WORKS



ADVANTAGES OF MVC

- Separation of Concerns:
  - Each component (Model, View, Controller) has a distinct role, promoting modularity and code organization.
- Reusability:
  - Components can be reused in different parts of the application or in other projects.
- Testability:
  - Components can be tested independently, facilitating unit testing.

ADVANTAGES OF MVC (CONT.)

- Flexibility & Maintainability:
  - Changes in one component do not directly affect the others, making the system more maintainable.
  - Bug fixes and errors are easier to identify because each section is isolated.
- Scalability:
  - The independent sections can be scaled up and made more complex without needing to change the other sections.

WHERE IT IS USED

- Web Development:
  - MVC is commonly used in web frameworks like ASP.NET MVC, Django, and Ruby on Rails.
- Desktop Applications:
  - GUI applications often employ MVC to separate UI logic (View) from data processing (Model) and user input handling (Controller).
- Mobile Development:
  - MVC principles are applied in mobile app development, with frameworks like Swift UI (iOS) and Jetpack (Android) using similar patterns.