MVC LECTURE

By Duncan

What is MVC

- MVC is an architectural pattern for implementing user interfaces, which stands for Model View Controller.

- It helps to separate the internal structure of data from the way data is shown to and received from the client.

- MVC software design allows for a clear "Separation of concerns" for a better division of labor and improved maintenance.

MODEL

- Manages information and the business logic.
- It represents the application data and behavior, in terms of its problem domain, and independent of the user interface.
- It does not communicate directly with the user.
- It dictates what data the app should contain.

VIEW

- Handles layout and display.

- It is visible to the user.

- Defines how the app data should be displayed.

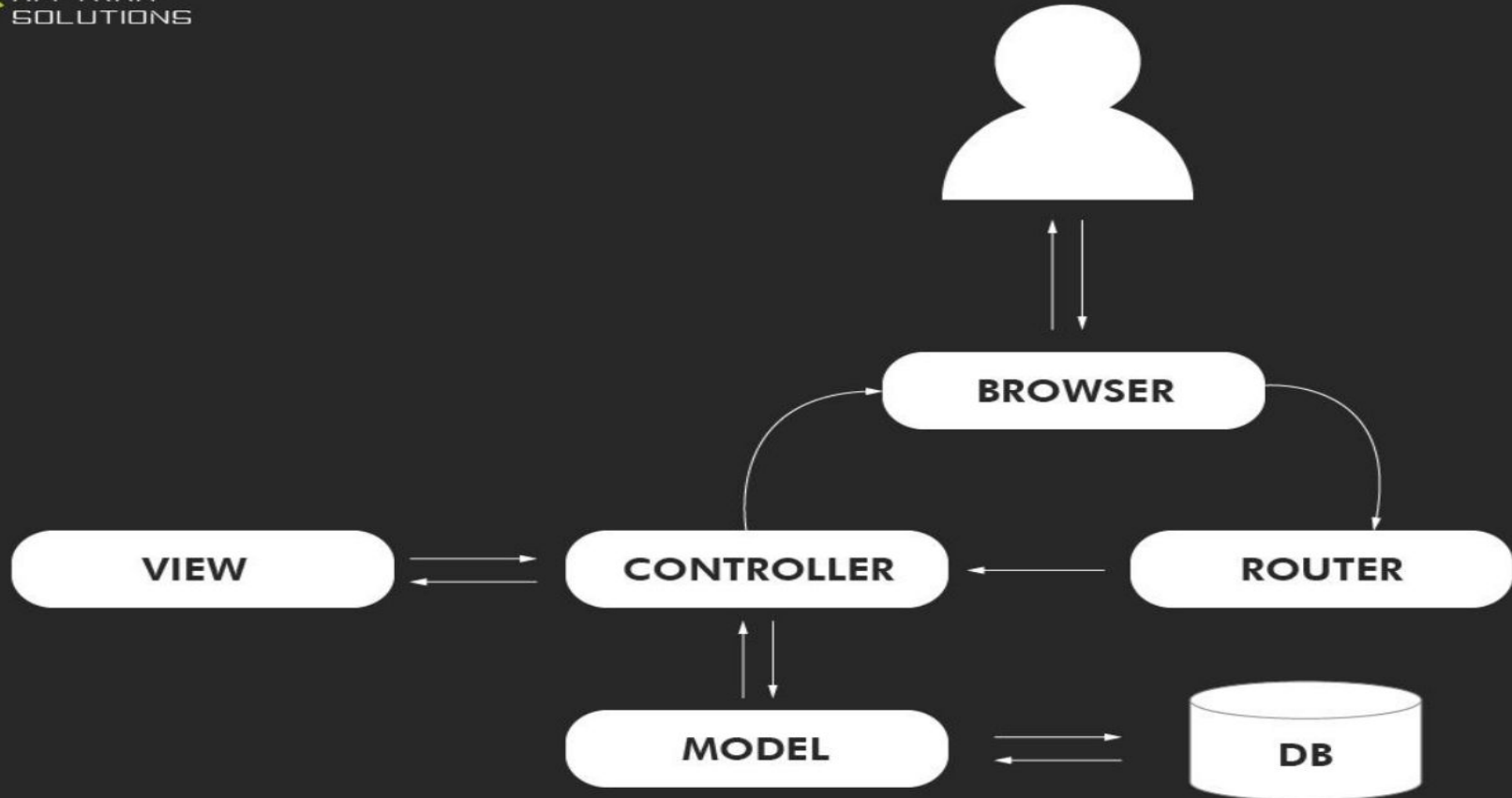- It allows for user interactions and displays the data/information

CONTROLLER

- It handles the application logic.
- It directs commands to/from the model and the view parts.
- It can communicate with both the Model and View.

The Web with MVC

- The data **MODEL** is contained in a database like MongoDB, MySQL, indexedDB, Oracle and other data storages.

- The **VIEW** user interface is written using templating languages such as **EJS**, handlebars and nunjucks.

- Javascript is used to implement the **CONTROLLER** and is responsible for handling an **HTTP** request.

MVC ARCHITECTURE

ROUTER

- **It selects the right controller to handle a request.**
- **Routing eliminates the need to map each URL with a physical file.**
- **Defines a URL pattern that maps to the request handler.**

MVC Architectural Pattern

Model	View	Controller	Router

Conclusion

- With MVC, each component has a distinct responsibility, and this results in better "separation of concerns", and a more maintainable application.

- When a request is sent, a controller is selected to handle this request. This controller will get the data from the database(MongoDB), put the data in the view and return the view to the client(Browser).

- The MVC architecture also has a router which has the responsibility of selecting the right controller or map URLs to the specified controller functions.