



WHY MVC?

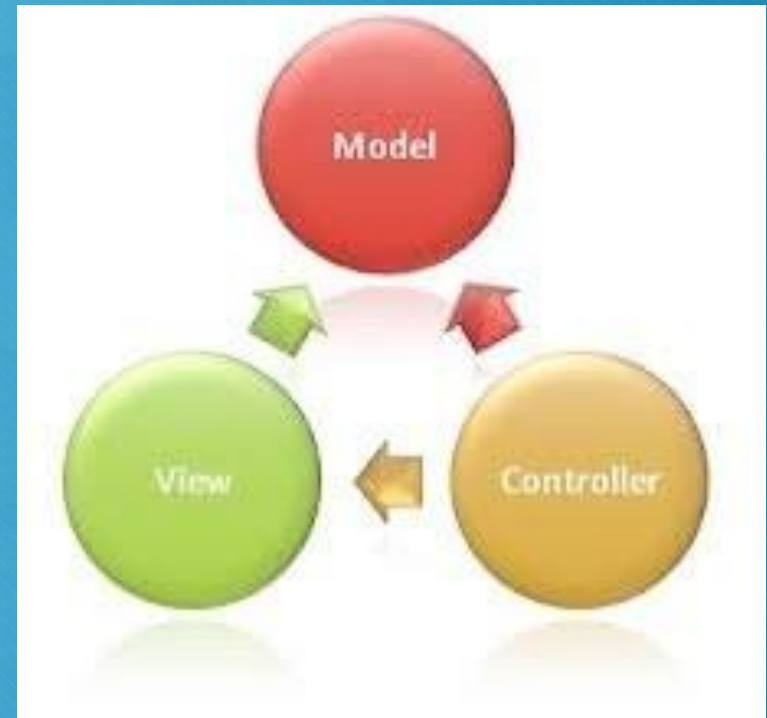
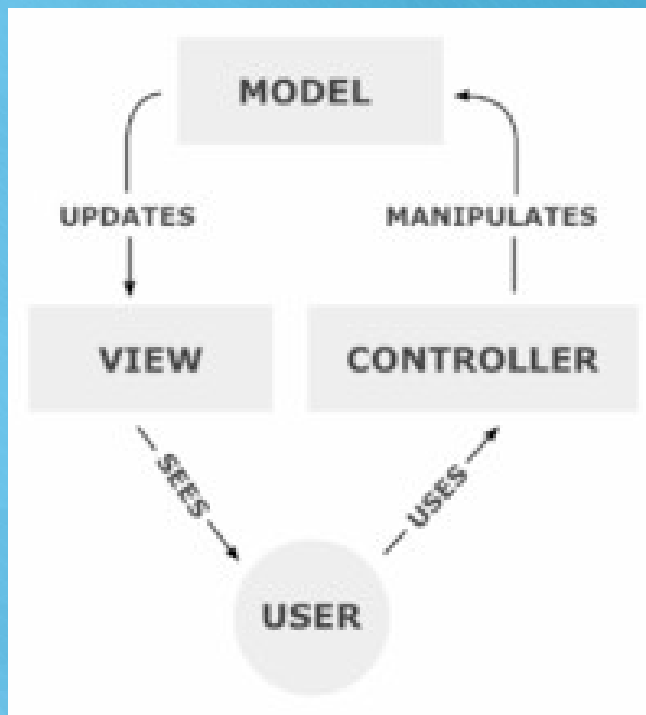
Arctech Info Private Limited



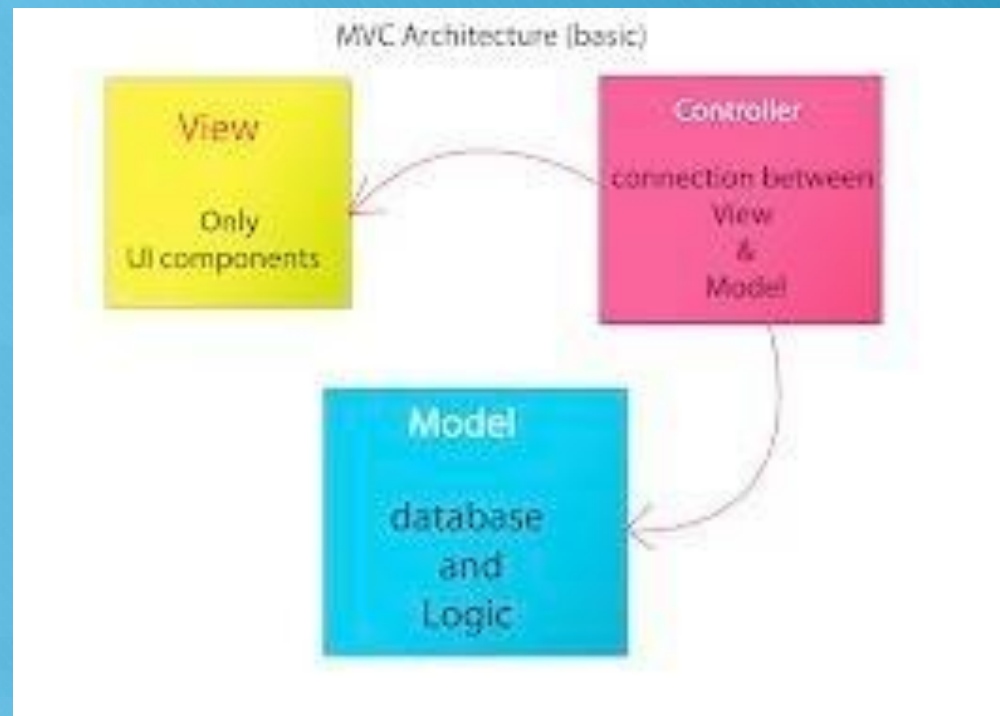
What is MVC?

- Model–view–controller
- A software architecture pattern which separates the representation of information from the user's interaction with it.
- The model consists of application data, business rules, logic, and functions.
- A view can be any output representation of data, such as a chart or a diagram. Multiple views of the same data are possible, such as a bar chart for management and a tabular view for accountants.

How MVC works!



How MVC Works | Page 2





Brief History Of MVC

- MVC was one of the first works to describe and implement software constructs in terms of their responsibilities.
- Trygve Reenskaug introduced MVC in the 1970s
- In the 1980s, Jim Althoff and others implemented a version of MVC .
- MVC was expressed as a general concept, in a 1988 article.



Brief History Of MVC- Page 2

- The Controller, in modern applications is a module, or an intermediary section of code, that mediates communication (between the Model and View) and unifies validation.
- Other aspects of the MVC also evolved, but as a variant of the original concept.
- HMVC, MVA, MVP, MVVM, and others that adapted MVC to different contexts.

Father of MVC

- Trygve Mikkjel Heyerdahl Reenskaug (born 1930) is a Norwegian computer scientist and professor emeritus of the University of Oslo.
- He formulated the model-view-controller (MVC) pattern for Graphic User Interface (GUI) software design in 1979 while visiting the Xerox Palo Alto Research Center (PARC).





Some MVC Frameworks

- ◊ ASP.NET (.NET)
- ◊ Django (Python)
- ◊ Rails(Ruby)
- ◊ Zend Framework (PHP)
- ◊ CodeIgniter (PHP)



WHY MVC?

- MVC (Model – View – Controller) is one of those concepts that only starts to make sense when you're dealing with larger applications and/or when working on the same application with several developers.
- If you've got 3 tasks that need to be done on a certain date, you won't feel the need to have some kind of task organization, but if you have to deal with dozens of different tasks, with various due dates you'll start using todo-lists.




WHY MVC-Page 2

- But when you start working with a team of people and you need to prioritize and sequence tasks, you realize simple to do-lists won't be enough.
- MVC's benefits are absolutely not restricted to large-scale, team-based development, but they become most obvious in such situations.



Example Project Case

- You made an web application/website for a flower shop owner with a small database table. it is a huge success and your client is extremely satisfied.
- They ask you to change the application, they want to use a different database and according to market demand they definitely need both iPhone and Android app as well.
- Now repeat this five times. The client keeps on asking modifications. expansions etc.
- These can be UI related changes and even complete backend architecture .

A stylized illustration at the top of the slide features a bright yellow sun with a small blue circle in the center, partially obscured by blue and white clouds. The background is a solid blue color.

YES , YOU will be in deep shit of pains.

- ◊ However if you used MVC from the start you'd notice that some things would've been less painful
- ◊ And more happier



Why?

- 90% of the code for the web application and the mobile app will be the same, but instead of saving the user data to a Shared Object or through a web service, you'd be using a local DB for instance.
- Without MVC chances are pretty high you'll be making modifications in a bunch of classes.
- The same applies to the UI for instance. Only the way it's presented to the user is different.



Advantages of Using MVC

- ◊ More Structural
- ◊ More Flexible
- ◊ Can maintain code in better ways.
- ◊ More suitable for future implementations
- ◊ Reduce Complexity



Disadvantages

- The complexity is high to develop the applications using this pattern without learning properly.
- Not right suitable for extra-small applications .
- The isolated development process by UI authors, business logic authors and controller authors may leads to delay in their respective modules development.

Practical Code Example

