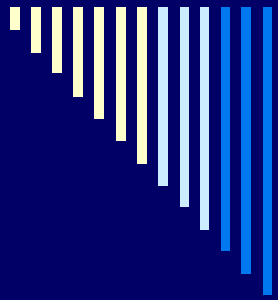


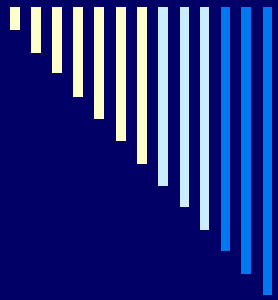
# ASP.NET MVC Framework

---



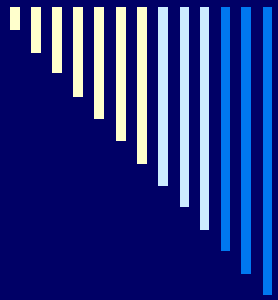
# WHAT IS MVC?

- A lightweight, highly testable presentation framework that is integrated w/existing ASP.NET.
- The Models Views Controllers (MVC) architectural pattern separates an application into three main components.



# MODELS

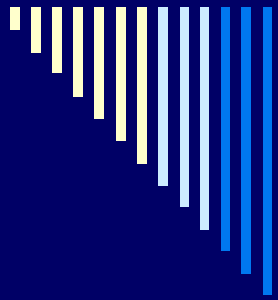
- Parts of the application that implements the domain logic. Which consists of:
    - Objects that could represent tables in our database that typically don't change and are the same across the board. Ex: The Job Seeker: SSN, birth date, phone number, mailing address, etc.
    - All business logic.
-



# VIEWS

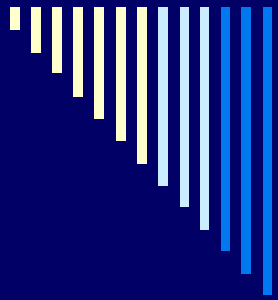
- Components that display the application's user interface (UI).

Example: All screens and forms. The browser.



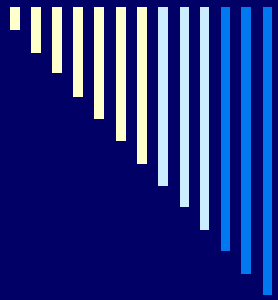
# CONTROLLERS

- Components that interpret the mouse and keyboard inputs from the user, informing the model and/or the view to change as appropriate. (The server side component handling the HTTP request)
- For example, the controller handles query-string values, and passes these values to the model, which in turn queries the db by using the values.



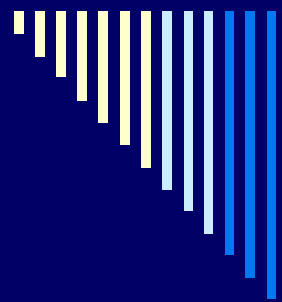
# BENEFITS

- **It is easier to manage complexity by dividing an application into the model, the view, and the controller.**
  - Doesn't use view state (think of it like form memory management) which makes the MVC framework ideal for developers who want full control over the behavior of an application.
  - Handles high productivity data scenarios.
  - Uses a Front Controller pattern that processes Web application requests through a single controller.
  - Provides greater support for test driven development (TDD).
  - Allows for integration with current software.
-



# MANAGING COMPLEXITY

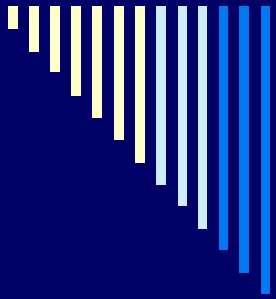
- Breaking down the application task into M-V-C allows us to better manage the complex relationships of Models (data structures and business logic), Views (the user interface), and Controllers (input logic).



# MANAGING COMPLEXITY

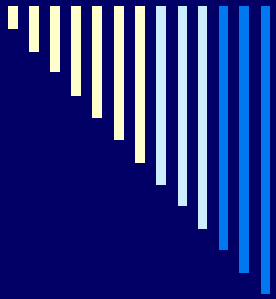
- It is highly extensible and pluggable. Everything in the MVC framework is designed so that it can be easily replaced/customized.

Note: User interface logic tends to change more frequently than business logic. The separation of presentation code and business logic allows us to minimize the introduction of errors that require the retesting of all business logic after minimal user interface changes because we would no longer have to modify the object directly just the view.



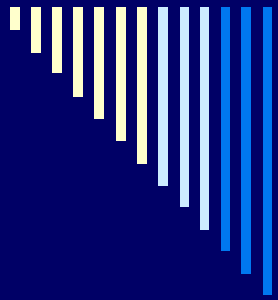
# BENEFITS

- It is easier to manage complexity by dividing an application into the model, the view, and the controller.
  - **Doesn't use view state (type of form memory management) which makes the MVC framework ideal for developers who want full control over the behavior of an application.**
  - Handles high productivity data scenarios.
  - Uses a Front Controller pattern that processes Web application requests through a single controller.
  - Provides greater support for test driven development (TDD).
  - Allows for integration with current software.
-



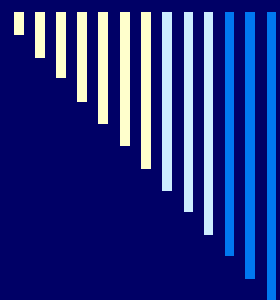
# BENEFITS

- It is easier to manage complexity by dividing an application into the model, the view, and the controller.
  - Doesn't use view state (type of form memory management) which makes the MVC framework ideal for developers who want full control over the behavior of an application.
  - **Handles high productivity data scenarios.**
  - Uses a Front Controller pattern that processes Web application requests through a single controller.
  - Provides greater support for test driven development (TDD).
  - Allows for integration with current software.
-



# BENEFITS

- It is easier to manage complexity by dividing an application into the model, the view, and the controller.
  - Doesn't use view state (type of form memory management) which makes the MVC framework ideal for developers who want full control over the behavior of an application.
  - Handles high productivity data scenarios.
  - **Uses a Front Controller pattern that processes Web application requests through a single controller.**
  - Provides greater support for test driven development (TDD).
  - Allows for integration with current software.
-

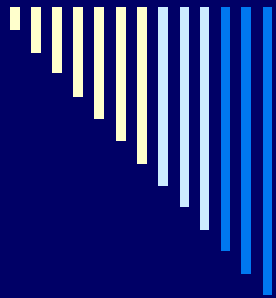


# URL MAPPING COMPONENT

- Includes a very powerful URL mapping component that enables you to build applications with clean URLs.

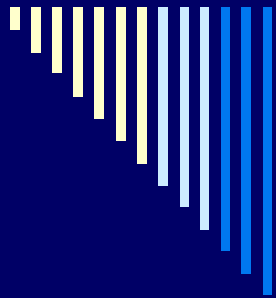
EXAMPLE: The 2 URLs listed below would be the same link thanks to M\$ MVC new mapping component.

- [https://www.kansasworks.com/ada/skillmatch/jobseeker\\_sm/jbs\\_clickres\\_dsp.cfm](https://www.kansasworks.com/ada/skillmatch/jobseeker_sm/jbs_clickres_dsp.cfm)
- <https://www.kansasworks.com/JobMatchProfile>



# BENEFITS

- It is easier to manage complexity by dividing an application into the model, the view, and the controller.
- Doesn't use view state (type of form memory management) which makes the MVC framework ideal for developers who want full control over the behavior of an application.
- Handles high productivity data scenarios.
- Uses a Front Controller pattern that processes Web application requests through a single controller.
- **Provides greater support for test driven development (TDD).**
- Allows for integration with current software.



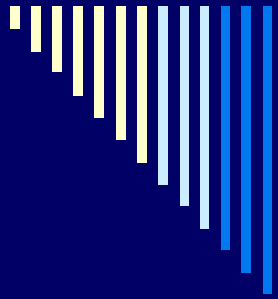
# TESTABILITY

- MVC enables clean separation of concerns, testability, and TDD (Test Driven Development) by default.
  - What is TDD?

Test Driven Development that implements automated unit testing, which defines and verifies the requirements of new code, first before you actually write the code itself.

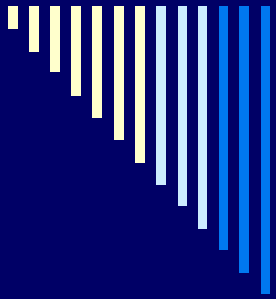
Note: You can test the application without having to run the Controllers within an ASP.NET process (making unit testing fast.) Currently using NUNIT for the unit testing framework.

---



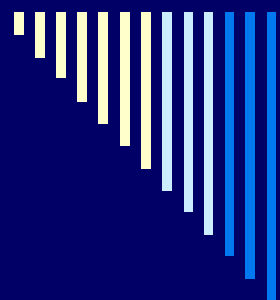
# TESTABILITY

- Creating automated tests for user interfaces is generally more difficult and time-consuming than for business logic. Therefore, reducing the amount of code that is directly tied to the user interface enhances the testability of the application.



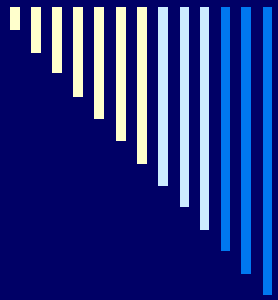
# BENEFITS

- It is easier to manage complexity by dividing an application into the model, the view, and the controller.
- Doesn't use view state (type of form memory management) which makes the MVC framework ideal for developers who want full control over the behavior of an application.
- Uses a Front Controller pattern that processes Web application requests through a single controller.
- Handles high productivity data scenarios.
- Provides greater support for test driven development (TDD).
- **Allows for integration with current software.**



# Supports Current Software

- The ASP.NET MVC Framework fully supports existing ASP.NET features.
  - Examples: forms/windows authentication, URL authorization, output and data caching, session/profile state management and health monitor.



---

# CONCLUSION

- Microsoft's newly recommended platform for enterprise level web applications is the ASP.NET MVC framework.
  - Although we anticipate the switch to slow things down in the short term it will speed development in the long term.
-