

- **Objective**

- To gain experience creating a simple app that uses PHP for server side scripting and database access.
- To gain experience using jQueryUI.

- **Description**

You will create a simple task manager that allows the user to add and delete several different types of tasks.

- **Instructions**

1. The task you create should look *something close to* the following:

## Task Manager

	Task	Type	Due Date
<input type="checkbox"/>	Grade Homework	NextAction	
<input type="checkbox"/>	Create report	WaitingFor: Bob	
<input type="checkbox"/>	Visit Paris	Someday/Maybe	
<input type="checkbox"/>	Discuss project	TalkTo: Susan	
<input type="checkbox"/>	Buy Mazarati	Future	
<input type="checkbox"/>	Do mailing	Waiting For: Mailing list	10/26/2012

Description:   
 Date:   
 Type:

The current list of tasks is shown in the table on the left. The form with 2 buttons on the right is used to add and delete tasks.

- To add a task the user does the following:
  - Fills in a description
  - Clicks on the date field and selects a date from the datepicker if desired
  - Selects one of 5 task types: **Next Action**, **Waiting For**, **Talk To**, **Future**, and **Someday/Maybe**.
  - If **Waiting For** or **Talk To** are selected, then a text box appears next to the drop down for the user to type in additional text.
  - Finally, the user clicks Add and the new task is saved to the database and appears in the task list.
- To delete one or more tasks, the user clicks the check boxes of the tasks to be deleted and clicks the Delete button. The tasks are immediately deleted from the database and the task list.
- Other details of the application function should be as described in class.

2. The application must use the MVC style architecture as exemplified by the Top40DemoMVC example. In fact you are welcome to start with the demo as your initial project. You are not welcome to leave the Top40 code in there – either clean it up or lose points.

3. The database should be initialized with the following sql script:

```
use taskmanager;
drop table if exists task;
create table task (
  id int NOT NULL AUTO_INCREMENT primary key,
  description varchar(30),
  type varchar(50),
  duedate varchar(10)
);
insert into task (description, type, duedate) values ('Buy milk', 'Location: Supermarket', '11/23/2014');
insert into task (description, type, duedate) values ('Grade Homework', 'NextAction', '');
insert into task (description, type, duedate) values ('Create report', 'WaitingFor: Bob', '');
insert into task (description, type, duedate) values ('Visit Paris', 'Someday/Maybe', '');
insert into task (description, type, duedate) values ('Discuss project', 'TalkTo: Susan', '');
insert into task (description, type, duedate) values ('Buy Mazarati', 'Future', '');
```

4. The project should contain the Action.php, Request.php, and index.php files from the Top40DemoMVC *with any necessary alterations*.

5. You need to create:

- (a) Action\_Add.php that contains the logic for executing the Add action.
- (b) Action\_Delete.php that contains the logic for executing the Delete action.

- (c) Action\_DisplayList.php that contains the logic for displaying the task manager page.
  - (d) Task.php that contains the definition of the Task class.
  - (e) tasklist.php that displays the task manager page.
  - (f) TaskManagerDAO.php that contains at least the following functions:
    - i. public function getTaskList()
    - ii. public function addTask(\$description, \$type, \$datedue, \$more)
    - iii. public function deleteTask(\$id)
6. The due date field should use the jQueryUI datepicker function to implement the calendar.
- (a) The tasklist.php file should include any necessary files.
  - (b) Create an element: `<input type="text" id="date" name="datedue" />` and then run `$("#input#date").datepicker();` when the page loads to initialize the date picker.
7. To submit your assignment, do a clean on your NetBeans project, ZIP it up and upload the zipped file to Coursesite. Place your NAME in a comment on every page in the application.