

# C#.NET CAPSTONE: TODO LIST REVISITED

NOTE: Points should be awarded for items that are written correctly in themselves, but don't actually work because other things are broken. There is a total of 10 points available for this lab.

**Intro:** Several months into your first developer position, Jill the project manager comes back to you. "Remember that task list we had you start as a console app a while back? We've finally decided--we want it as a web app. Think you can dust it off and make it for us? This time we only want users to see their own tasks, but it should basically work the same as the console app."

**Task:** Create a web-based task manager

## What Will the Application Do?

- **1 Point:** A user can create a new account with an email address and password
- **1 Point:** Once logged in, the user can create a new `todoItem`. Todo Items consist of:
  - The ID of the user who owns it (user doesn't get to edit this field)
  - A todo item description
  - The due date
  - Whether it's complete or not
- **1 Point:** The user should see a list of all todo items they own
  - **1 Point:** There should be a button or checkbox to mark the todo item complete
  - **1 Point:** There should be a button to delete the todo item

## Build Specifications

1. **1 Point:** Build this as an MVC web application
2. **1 Point:** Store users and tasks in separate SQL database tables
3. **1 Point:** Minimum views: Welcome/login page, todo item list, add task view

## Hints

- When making the SQL call for which tasks to display, show only the todo items which have this user's ID

## Additional Requirements

- **-2 Points:** If there are any syntax errors or if the program does not run.

## Extended Challenges (2 Points Maximum)

- **1 Point:** Let the user search for a todo item by a word or words in the description
- **1 Point:** Let the user sort or filter by due date
- **1 Point:** Let the user sort or filter by completion status
- **1 Point:** Complete the user Login using Identity