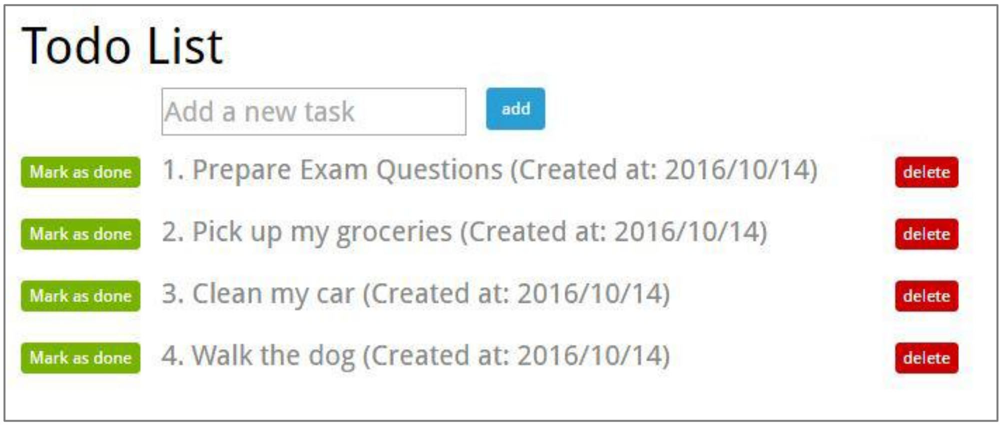
CS572 - Programming Question Practice

Complete the code for the following Express Todo App:

The main page will show all incomplete tasks, with a form on top to add a new task. Every task can be marked as done, once it's marked as done it will only be displayed in the completed lists page. You may delete any task by clicking on the delete button. You may assume your document structure to be:

```
tasks { _id: ObjectId(), task: "", status: "", created_date: "",
                              updated_date: ""}
```

You may **assume** that on the client-side, clicking on the (Mark as done) button or (delete) button will trigger an AJAX calls with the appropriate method to your server. All you need to do is to take care of receiving the request and write the appropriate code to serve all the routes below (Backend ONLY)



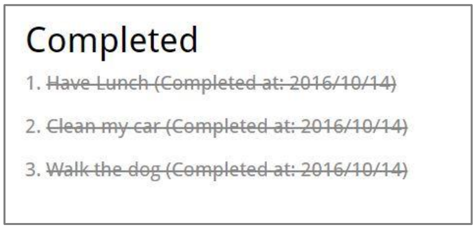
Todo List

1. Prepare Exam Questions (Created at: 2016/10/14)
2. Pick up my groceries (Created at: 2016/10/14)
3. Clean my car (Created at: 2016/10/14)
4. Walk the dog (Created at: 2016/10/14)

The app RESTful routes mapping:

GET	/tasks	Display all incomplete todo tasks
POST	/tasks	Add new Task
POST	/tasks/:task_id	Mark Task as completed
DELETE	/tasks/:task_id	Delete Task
GET	/tasks/completed	Display completed tasks

A completed tasks list would look like this:



Completed

1. ~~Have Lunch (Completed at: 2016/10/14)~~
2. ~~Clean my car (Completed at: 2016/10/14)~~
3. ~~Walk the dog (Completed at: 2016/10/14)~~

You may assume that the views folder contains two files: `tasks.jade` and `tasks_completed.jade`, you don't need to write any code for these files.

You are required to use Express Router and complete the code in both `app.js` and `/routes/tasks.js` files.

There is no need to handle any input validation or add any security configuration in your application (You might be asked to do that in real exam)

Notice that there is a variable `db` and `collection` initiated for you, which contains a reference to the collection in the current active connection of MongoDB Server.

app.js

```
var express = require('express');
var path = require('path');
var cookieParser = require('cookie-parser');
var bodyParser = require('body-parser');
var db = require('mongoskin').db('mongodb://localhost:27017/todo');
var collection = db.tasks;
var tasks = require('./routes/tasks');

var app = express();

app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));









app.use('/tasks', tasks);

app.use(function(err, req, res, next) {
  res.status(err.status || 500);
  res.render('error', {
    message: err.message,
    error: {}
  });
});
app.listen(3000);
```

/routes/tasks.js