

Unit 6

Exercise - Create a basic web application

Create the Books web application

Create the files

З Shell я запускаю ці команди, щоб створити папки та файли для мого веб-додатку:

```
stideshowbobgot@localhost:~/university/infrastructure/Lab1/5_BuildAndRunAWebApplication/Unit6$ ssh -l id_rsa azureuser@52.228.22.44
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1054-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Mon Feb 26 20:46:01 UTC 2024

System load:  0.0          Processes:      112
Usage of /:   9.4% of 28.89GB Users logged in: 0
Memory usage: 13%         IPv4 address for eth0: 10.0.0.4
Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Mon Feb 26 20:08:33 2024 from 91.211.120.221
azureuser@PanchenkoSerhiStack:~$
```

```
azureuser@PanchenkoSerhiStack:~$ cd ~
azureuser@PanchenkoSerhiStack:~$ mkdir Books
azureuser@PanchenkoSerhiStack:~$ touch Books/server.js
azureuser@PanchenkoSerhiStack:~$ touch Books/package.json
azureuser@PanchenkoSerhiStack:~$ mkdir Books/app
azureuser@PanchenkoSerhiStack:~$ touch Books/app/model.js
azureuser@PanchenkoSerhiStack:~$ touch Books/app/routes.js
azureuser@PanchenkoSerhiStack:~$ mkdir Books/public
azureuser@PanchenkoSerhiStack:~$ touch Books/public/script.js
azureuser@PanchenkoSerhiStack:~$ touch Books/public/index.html
azureuser@PanchenkoSerhiStack:~$
```

Create the data model

З редактора я відкриваю app/model.js і додаю наступне:

```
azureuser@PanchenkoSerhiiStack: ~/Books
sideshowbobgot@localhost: ~ × azureuser@PanchenkoSerhiiStack... × side

GNU nano 4.8 app/model.js
var mongoose = require('mongoose');
var dbHost = 'mongodb://localhost:27017/Books';
mongoose.connect(dbHost, { useNewUrlParser: true } );
mongoose.connection;
mongoose.set('debug', true);
var bookSchema = mongoose.Schema( {
  name: String,
  isbn: {type: String, index: true},
  author: String,
  pages: Number
});
var Book = mongoose.model('Book', bookSchema);
module.exports = Book;
```

Create the Express.js routes that handle HTTP requests

З редактора я відкриваю app/routes.js і додаю наступний код:

```
azureuser@PanchenkoSerhiiStack: ~/Books
sideshowbobgot@localhost: ~ × azureuser@PanchenkoSerhiiStack... × side

GNU nano 4.8 app/routes.js
var path = require('path');
var Book = require('./model');
var routes = function(app) {
  app.get('/book', function(req, res) {
    Book.find({}, function(err, result) {
      if ( err ) throw err;
      res.json(result);
    });
  });
  app.post('/book', function(req, res) {
    var book = new Book( {
      name:req.body.name,
      isbn:req.body.isbn,
      author:req.body.author,
      pages:req.body.pages
    });
    book.save(function(err, result) {
      if ( err ) throw err;
      res.json( {
        message:"Successfully added book",
        book:result
      });
    });
  });
  app.delete("/book/:isbn", function(req, res) {
    Book.findOneAndRemove(req.query, function(err, result) {
      if ( err ) throw err;
      res.json( {
        message: "Successfully deleted the book",
        book: result
      });
    });
  });
  app.get('*', function(req, res) {
    res.sendFile(path.join(__dirname + '/public', 'index.html'));
  });
};
module.exports = routes;
```

Create the client-side JavaScript application

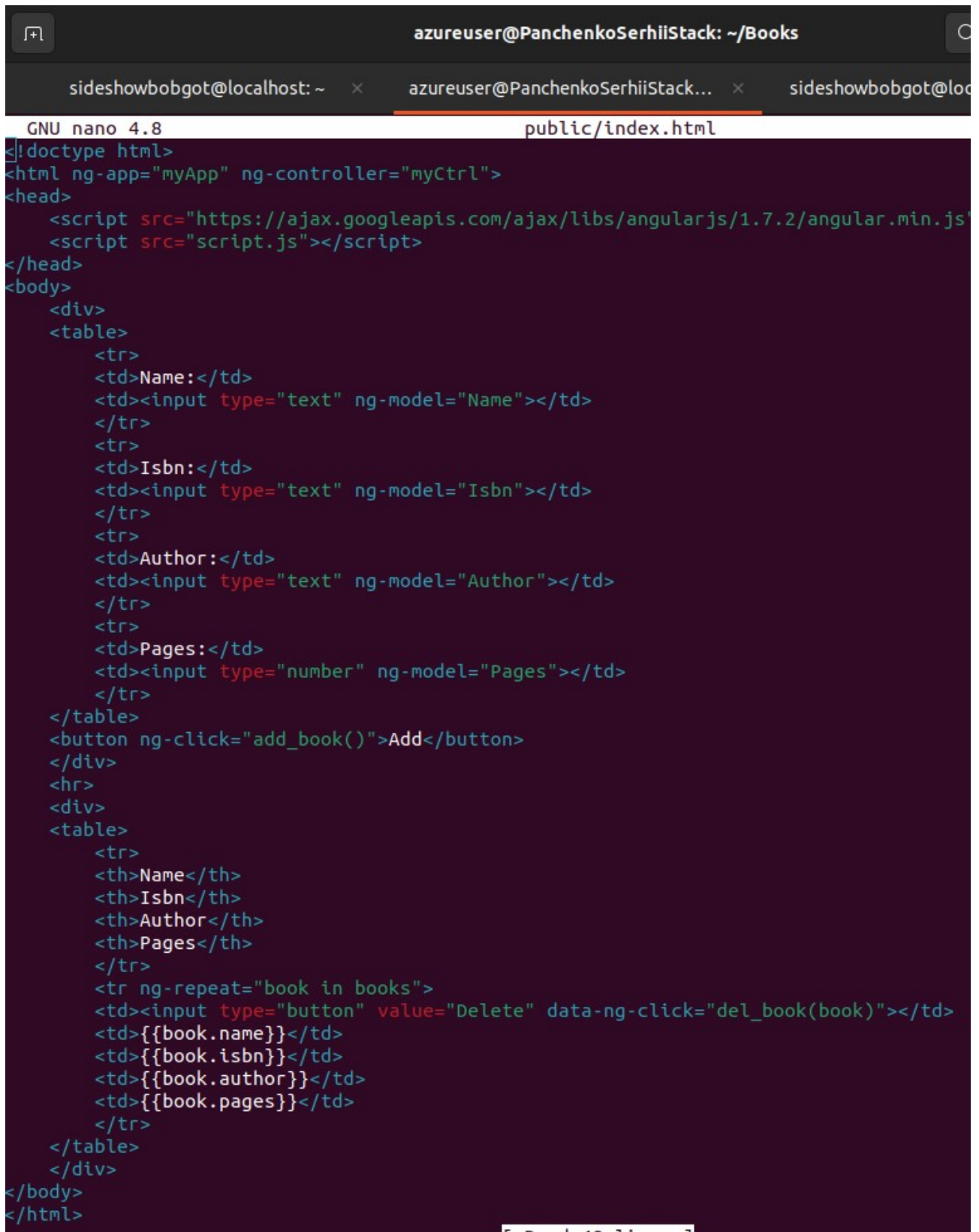
З редактора я відкриваю public/script.js і додаю цей код:



```
GNU nano 4.8 public/script.js
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope, $http) {
  var getData = function() {
    return $http( {
      method: 'GET',
      url: '/book'
    }).then(function successCallback(response) {
      $scope.books = response.data;
    }, function errorCallback(response) {
      console.log('Error: ' + response);
    });
  };
  getData();
  $scope.del_book = function(book) {
    $http( {
      method: 'DELETE',
      url: '/book/:isbn',
      params: {'isbn': book.isbn}
    }).then(function successCallback(response) {
      console.log(response);
      return getData();
    }, function errorCallback(response) {
      console.log('Error: ' + response);
    });
  };
  $scope.add_book = function() {
    var body = '{ "name": "' + $scope.Name +
    '" , "isbn": "' + $scope.Isbn +
    '" , "author": "' + $scope.Author +
    '" , "pages": "' + $scope.Pages + '" }';
    $http({
      method: 'POST',
      url: '/book',
      data: body
    }).then(function successCallback(response) {
      console.log(response);
      return getData();
    }, function errorCallback(response) {
      console.log('Error: ' + response);
    });
  };
});
```

Create the user interface

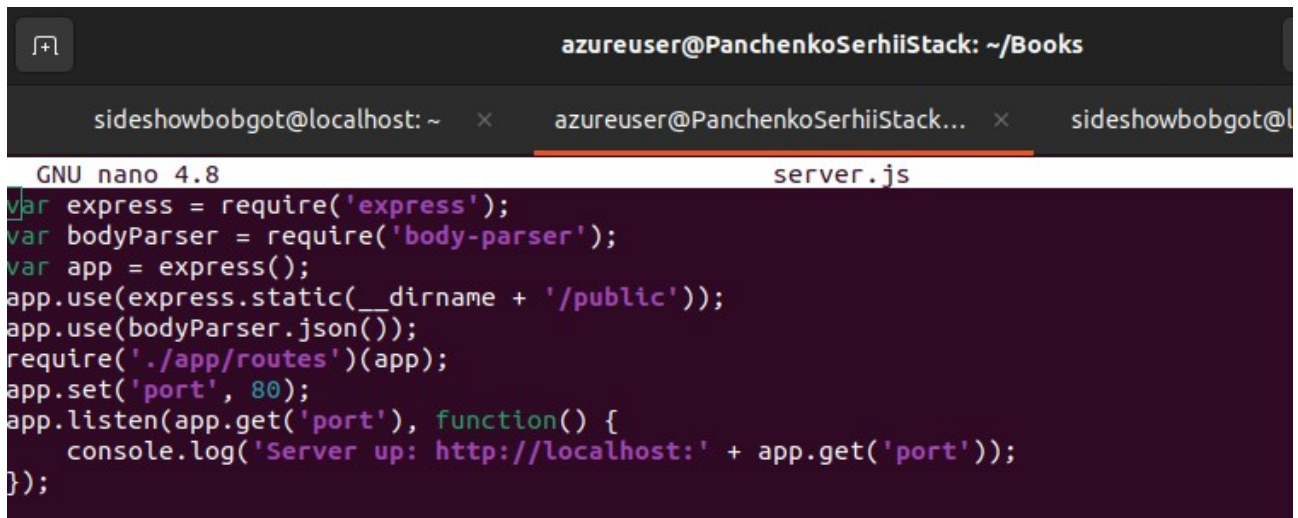
З редактора я відкриваю public/index.html і додаю цей код:



```
GNU nano 4.8 public/index.html
<!doctype html>
<html ng-app="myApp" ng-controller="myCtrl">
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.2/angular.min.js">
  <script src="script.js"></script>
</head>
<body>
  <div>
    <table>
      <tr>
        <td>Name:</td>
        <td><input type="text" ng-model="Name"></td>
      </tr>
      <tr>
        <td>Isbn:</td>
        <td><input type="text" ng-model="Isbn"></td>
      </tr>
      <tr>
        <td>Author:</td>
        <td><input type="text" ng-model="Author"></td>
      </tr>
      <tr>
        <td>Pages:</td>
        <td><input type="number" ng-model="Pages"></td>
      </tr>
    </table>
    <button ng-click="add_book()">Add</button>
  </div>
  <hr>
  <div>
    <table>
      <tr>
        <th>Name</th>
        <th>Isbn</th>
        <th>Author</th>
        <th>Pages</th>
      </tr>
      <tr ng-repeat="book in books">
        <td><input type="button" value="Delete" data-ng-click="del_book(book)"></td>
        <td>{{book.name}}</td>
        <td>{{book.isbn}}</td>
        <td>{{book.author}}</td>
        <td>{{book.pages}}</td>
      </tr>
    </table>
  </div>
</body>
</html>
```


Create the Express.js server to host the application

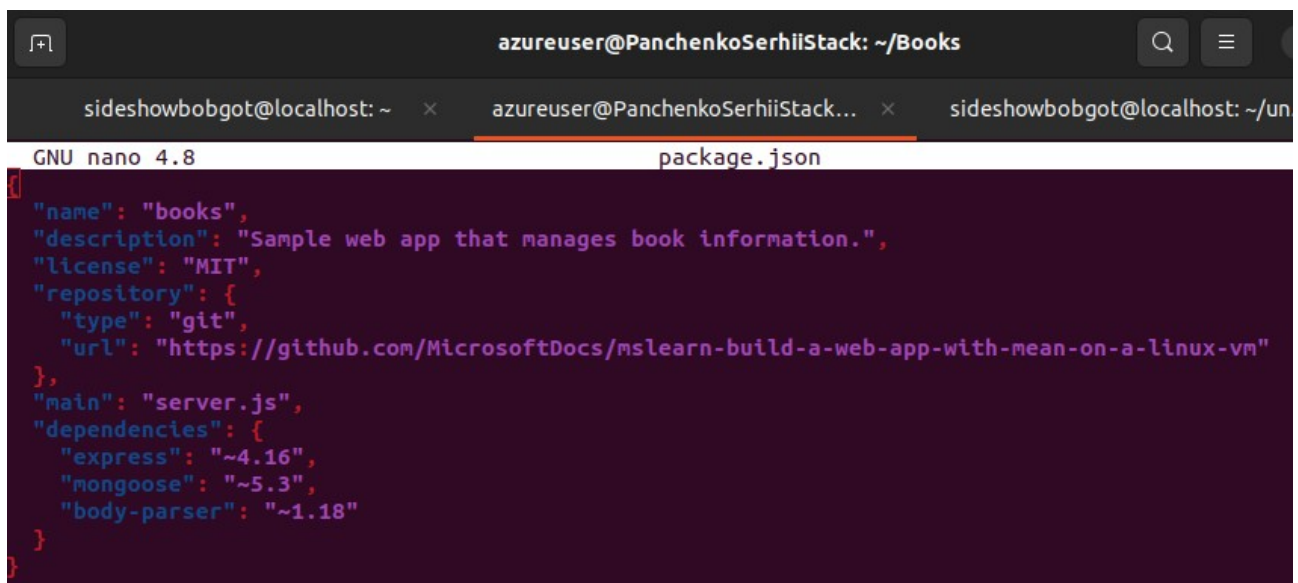
З редактора я відкриваю server.js і додаю цей код:

A screenshot of a terminal window with a dark background. The title bar shows 'azureuser@PanchenkoSerhiiStack: ~/Books'. There are three tabs: 'sideshowbobgot@localhost: ~', 'azureuser@PanchenkoSerhiiStack...', and 'sideshowbobgot@l...'. The terminal prompt is 'GNU nano 4.8' and the file being edited is 'server.js'. The code is as follows:

```
var express = require('express');
var bodyParser = require('body-parser');
var app = express();
app.use(express.static(__dirname + '/public'));
app.use(bodyParser.json());
require('./app/routes')(app);
app.set('port', 80);
app.listen(app.get('port'), function() {
  console.log('Server up: http://localhost:' + app.get('port'));
});
```

Define package information and dependencies

Пам'ятаю, що package.json надає інформацію про мій додаток, включаючи його назву, опис та які пакети Node.js необхідні для його роботи. З редактора я відкриваю package.json і додаю цей код:

A screenshot of a terminal window with a dark background. The title bar shows 'azureuser@PanchenkoSerhiiStack: ~/Books'. There are three tabs: 'sideshowbobgot@localhost: ~', 'azureuser@PanchenkoSerhiiStack...', and 'sideshowbobgot@localhost: ~/un...'. The terminal prompt is 'GNU nano 4.8' and the file being edited is 'package.json'. The code is as follows:

```
{
  "name": "books",
  "description": "Sample web app that manages book information.",
  "license": "MIT",
  "repository": {
    "type": "git",
    "url": "https://github.com/MicrosoftDocs/mslearn-build-a-web-app-with-mean-on-a-linux-vm"
  },
  "main": "server.js",
  "dependencies": {
    "express": "~4.16",
    "mongoose": "~5.3",
    "body-parser": "~1.18"
  }
}
```

Install additional Node packages

Припустимо, що під час процесу розробки я визначив додаткові пакети Node, які хочу використовувати. Наприклад, пам'ятаю, що app/model.js починається з цього рядка. Виконую npm install, щоб встановити залежні пакети:

```
azureuser@PanchenkoSerhiStack:~/Books$ npm install
added 69 packages, and audited 70 packages in 4s
8 vulnerabilities (2 moderate, 5 high, 1 critical)
To address all issues, run:
  npm audit fix --force
Run `npm audit` for details.
npm notice
npm notice New major version of npm available! 8.19.4 -> 10.4.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.4.0
npm notice Run npm install -g npm@10.4.0 to update!
npm notice
azureuser@PanchenkoSerhiStack:~/Books$
```

Test the application

Тепер я готовий протестувати свій веб-додаток на Node.js! З директорії ~/Books я запускаю цю команду, щоб стартувати веб-додаток: Ця команда запускає додаток, слухаючи порт 80 для вхідних HTTP запитів. З окремої вкладки браузера я переходжу до публічної IP-адреси моєї ВМ. Я бачу головну сторінку, яка включає веб-форму.

```
azureuser@PanchenkoSerhiStack:~/Books$ sudo node server.js
(node:23297) Warning: Accessing non-existent property 'count' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:23297) Warning: Accessing non-existent property 'findOne' of module exports inside circular dependency
(node:23297) Warning: Accessing non-existent property 'remove' of module exports inside circular dependency
(node:23297) Warning: Accessing non-existent property 'updateOne' of module exports inside circular dependency
Server up: http://localhost:80
Mongoose: books.ensureIndex({ isbn: 1 }, { background: true })
(node:23297) DeprecationWarning: collection.ensureIndex is deprecated. Use createIndexes instead.

```

Name:

Isbn:

Author:

Pages: