

التاريخ: 2024-06-19.



جامعة طرابلس / كلية تقنية المعلومات قسم هندسة البرمجيات.



السيد المحترم / د. عبدالحميد الواعر.

الموضوع:- تقرير عن WAFs البنية على نمط MVC.

بعد التحية والسلام.

في إطار مادة إعادة استخدام البرمجيات (ITSE422)، يطلب إنشاء تطبيق ويب باستخدام إحدى WAFs البنية على MVC pattern، وكتاب تقرير يشرح الـ MVC pattern وكيف سيتم استخدامه عند إنشاء التطبيق. عليه تقدم الى حضرتكم هذا التقرير.

مع خالص التحية،

الطالب: أحمد مصطفى المحجوب.

رقم القيد: 2191804140.

الطالب: علي محمد الغناي.

رقم القيد: 2191804737.

الطالب: انس فتحي المغربي.

رقم القيد: 2191804670.

مقدمة

يوضح هذا التقرير تطوير تطبيق ويب بسيط لإدارة وعرض معلومات الطلاب. يتم بناء التطبيق باستخدام نمط MVC (Model-View-Controller) ومجموعة من إطارات تطبيقات الويب (WAFs). الهدف الرئيسي هو فهم وتنفيذ نمط MVC لتحسين إعادة استخدام البرمجيات وتنظيمها.

الأهداف

1. تطوير تطبيق ويب لعرض وإدارة معلومات الطلاب (الاسم، العنوان، العمر).
2. تنفيذ نمط MVC لتنظيم التطبيق.
3. تقديم تقرير مفصل يشرح نمط MVC واستخدامه في تطبيق الويب.

التقنيات المستخدمة

- **Node.js**: بيئة تشغيل JavaScript لتطوير الخادم.
- **Express**: إطار عمل ويب لبناء التطبيق.
- **Pug**: محرك القوالب لعرض النماذج.
- **Sequelize**: ORM للتعامل مع قاعدة البيانات.
- **SQLite**: قاعدة بيانات لتخزين معلومات الطلاب.
- **Axios**: عميل HTTP لإجراء طلبات API.
- **Nodemon**: أداة لإعادة تشغيل الخادم تلقائيًا عند تغيير الملفات.

هيكل المشروع:

```
root:
  |_ controllers
    |_ dashboard.js
    |_ delete-all-student.js
    |_ student.js
  |_ models
    |_ Student.js
  |_ public/js
    |_ dashboard.js
  |_ routes
    |_ dashboard.js
    |_ delete-all-student.js
    |_ add-student.js
  |_ views
    |_ dashboard.pug
    |_ add-student.pug
    |_ layout.pug
  |_ db.js
  |_ mvc_demo.sqlite
  |_ server.js
```

التاريخ: 2024-06-19.

شرح نمط MVC

يقسم نمط MVC التطبيق إلى ثلاث مكونات مترابطة:

1. **النموذج (Model):** يدير البيانات والمنطق التجاري.
2. **العرض (View):** يتولى عرض المعلومات.
3. **التحكم (Controller):** يدير منطق التطبيق وإدخال المستخدم.

يساعد هذا التقسيم في تنظيم قاعدة الكود وتحسين قابلية التوسع وتعزيز إعادة استخدام الكود.

شرح طريقة تنفيذ البرنامج (Implementation):

النموذج (Model):

يعرّف نموذج **Student.js** مخطط الطالب ويتفاعل مع قاعدة البيانات باستخدام **SQLite** باستخدام **Sequelize**.

```
JS Student.js ×
models > JS Student.js > ...
1  const { Sequelize, DataTypes } = require("sequelize");
2  const sequelize = require("../db");
3
4  module.exports = sequelize.define("Student", {
5    id: {
6      type: DataTypes.INTEGER,
7      autoIncrement: true,
8      primaryKey: true,
9    },
10   name: {
11     type: DataTypes.STRING,
12   },
13   address: {
14     type: DataTypes.STRING,
15   },
16   age: {
17     type: DataTypes.INTEGER,
18   },
19 });
20
```

التاريخ: 2024-06-19.

العرض (View)

تم تعريف العروض باستخدام قوالب Pug. تقوم بتقديم معلومات الطالب وتوفر واجهات لإضافة وحذف الطلاب.

:Dashboard.pug

```
dashboard.pug ×
views > dashboard.pug
1 extends layout.pug
2
3 block content
4   .container
5     h4 Dashboard
6     .mb-3
7       a.btn.btn-primary(href='/add-student') Add New Student
8       button.btn.btn-danger.ml-3(onclick='deleteAllStudents()') Delete All Students
9
10    if students.length > 0
11      each student in students
12        .card.mb-3
13          .card-body
14            h5.card-title= student.name
15            p.card-text
16              | Age: #{student.age}
17            p.card-text
18              | Address: #{student.address}
19    else
20      p No students found.
21
22    script(src='/js/dashboard.js')
23
```

التاريخ: 2024-06-19.

:Add-student.pug

```
add-student.pug ×
views > add-student.pug
1 extends layout.pug
2 block content
3   .container
4     .row.justify-content-center
5       .col-md-8
6         if error
7           .alert.alert-warning= error
8         .card
9           .card-header Add Student To DataBase
10          .card-body
11            form.form-horizontal(method='post' action='/add-student')
12              .mb-3
13                label.form-label.control-label(for='name') Name
14                input#name.form-control(type='text' name='name')
15              .mb-3
16                label.form-label.control-label(for='address') Address
17                input#address.form-control(type='text' name='address')
18              .mb-3
19                label.form-label.control-label(for='age') Age
20                input#age.form-control(type='number' name='age')
21              .mb-3.py-3
22                button.btn.btn-primary.login-button(type='submit') Save Student
23
```

:Layout.pug

```
layout.pug ×
views > layout.pug
1 doctype html
2 head
3   meta(charset='utf-8')
4   title Reuse MVC Project
5   meta(name='viewport' content='width=device-width, initial-scale=1')
6   link(href='https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css' rel='stylesheet' integrity='sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2drRjJXh0JhY66WwALEwIh' crossorigin='anonymous')
7   nav.navbar.navbar-expand-lg.navbar-dark.bg-dark
8     .container-fluid
9       a.navbar-brand(href='#') Reuse MVC Project
10      button.navbar-toggler(type='button' data-bs-toggle='collapse' data-bs-target='#navbarSupportedContent' aria-controls='navbarSupportedContent' aria-expanded='false' aria-label='Toggle navigation')
11      span.navbar-toggler-icon
12      if name
13        #navbarSupportedContent.collapse.navbar-collapse.justify-content-end
14          form.d-flex
15            span.navbar-text.text-light.pe-2 Hello #name
16            a.btn.btn-outline-light(href='/logout') Logout
17      .py-2
18      block content
19      script(src='https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js')
20      script(src='https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js' integrity='sha384-Mrcw6ZMFYlzcLAbnLNUtUvF8sA7MsXsP1Uy30M4YLEwSfAP+JcXn/tWtIaxVM' crossorigin='anonymous')
```

التاريخ: 2024-06-19.

التحكم (Controller)

المتحكمات تدير منطق التطبيق وإدخال المستخدم، وتدير تدفق البيانات بين النماذج والعروض.

:student.js

```
JS student.js ×
controllers > JS student.js > ...
1  const Student = require("../models/Student");
2  ⚡
3  module.exports = {
4    addStudentView: (req, res) => {
5      res.render("add-student");
6    },
7    addStudent: async (req, res) => {
8      const { name, address, age } = req.body;
9      if (!name || !address || !age) {
10       return res.render("add-student", { error: "Please fill all fields" });
11     }
12     await Student.create({ name, address, age });
13
14     return res.redirect("/");
15   },
16 };
17
```

التاريخ: 2024-06-19.

:dashboard.js

```
JS dashboard.js ×
controllers > JS dashboard.js > [?] <unknown> > dashboardView
1  const Student = require("../models/Student");
2
3  module.exports = {
4    dashboardView: async (req, res) => {
5      try {
6        const students = await Student.findAll();
7        res.render("dashboard", { students });
8      } catch (error) {
9        res.render("dashboard", {
10          students: [],
11          error: "Failed to load students",
12        });
13      }
14    },
15  };
16
```


التاريخ: 2024-06-19.

delete-all-student.js:

```
JS delete-all-student.js ×
controllers > JS delete-all-student.js > ...
1  const Student = require("../models/Student");
2
3  module.exports = {
4    deleteAllStudents: async (req, res) => {
5      try {
6        await Student.truncate();
7        res.json({ success: true });
8      } catch (error) {
9        console.log(error);
10       res.json({ success: false, error: "Failed to delete students" });
11     }
12   },
13 };
14
```

المسارات (Routes):

المسارات تعرف نقاط النهاية وتعينها لدوال المتحكمات المناسبة.

routes/dashboard.js:

```
JS dashboard.js ×
routes > JS dashboard.js > ...
1  const express = require("express");
2  const dashboardController = require("../controllers/dashboard");
3
4  const router = express.Router();
5  router.get("/", dashboardController.dashboardView);
6
7  module.exports = router;
```

routes/add-student.js:

التاريخ: 2024-06-19.

```
JS add-student.js ×
routes > JS add-student.js > ...
1  const express = require("express");
2  const studentController = require("../controllers/student");
3
4  const router = express.Router();
5  router.get("/", studentController.addStudentView);
6  router.post("/", studentController.addStudent);
7
8  module.exports = router;
9
```

routes/delete-all-student.js:

```
JS delete-all-student.js ×
routes > JS delete-all-student.js > ...
1  const express = require("express");
2  const deleteAllStudents = require("../controllers/delete-all-student");
3
4  const router = express.Router();
5  router.post("/", deleteAllStudents.deleteAllStudents);
6
7  module.exports = router;
8
```

الاتصال بقاعدة البيانات

db.js:

التاريخ: 2024-06-19.

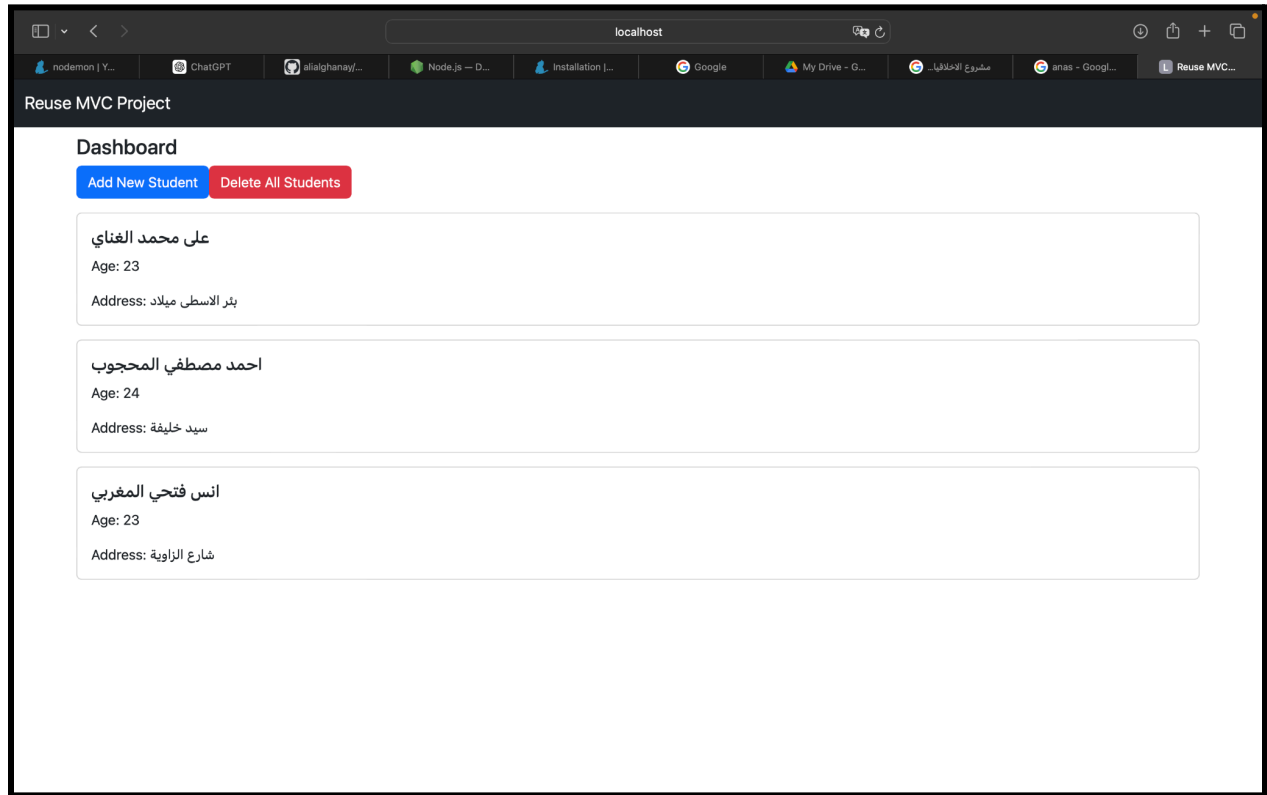
```
JS db.js ×
JS db.js > ...
1  const path = require("path");
2  const { Sequelize } = require("sequelize");
3
4  module.exports = new Sequelize({
5    dialect: "sqlite",
6    storage: path.join(__dirname, "mvc_demo.sqlite"),
7  });
8
```

إعداد الخادم
server.js:

```
JS server.js ×
JS server.js > ...
1  const express = require("express");
2  const app = express();
3  const PORT = 8080;
4  const bodyParser = require("body-parser");
5  const path = require("path");
6  const db = require("./db.js");
7  const dashboardRoutes = require("./routes/dashboard");
8  const addStudent = require("./routes/add-student");
9  const deleteAll = require("./routes/delete-all-student.js");
10
11  app.use(bodyParser.urlencoded({ extended: true }));
12  app.use(bodyParser.json());
13
14  app.set("view engine", "pug");
15  app.use(express.static(path.join(__dirname, "public")));
16
17  app.use("/", dashboardRoutes);
18  app.use("/add-student", addStudent);
19  app.use("/delete-all", deleteAll);
20
21  db.sync({ force: false }).then(() => {
22    app.listen(PORT, console.log("Server is running on port: " + PORT));
23  });
24
```

التاريخ: 2024-06-19.

النتيجة النهائية:

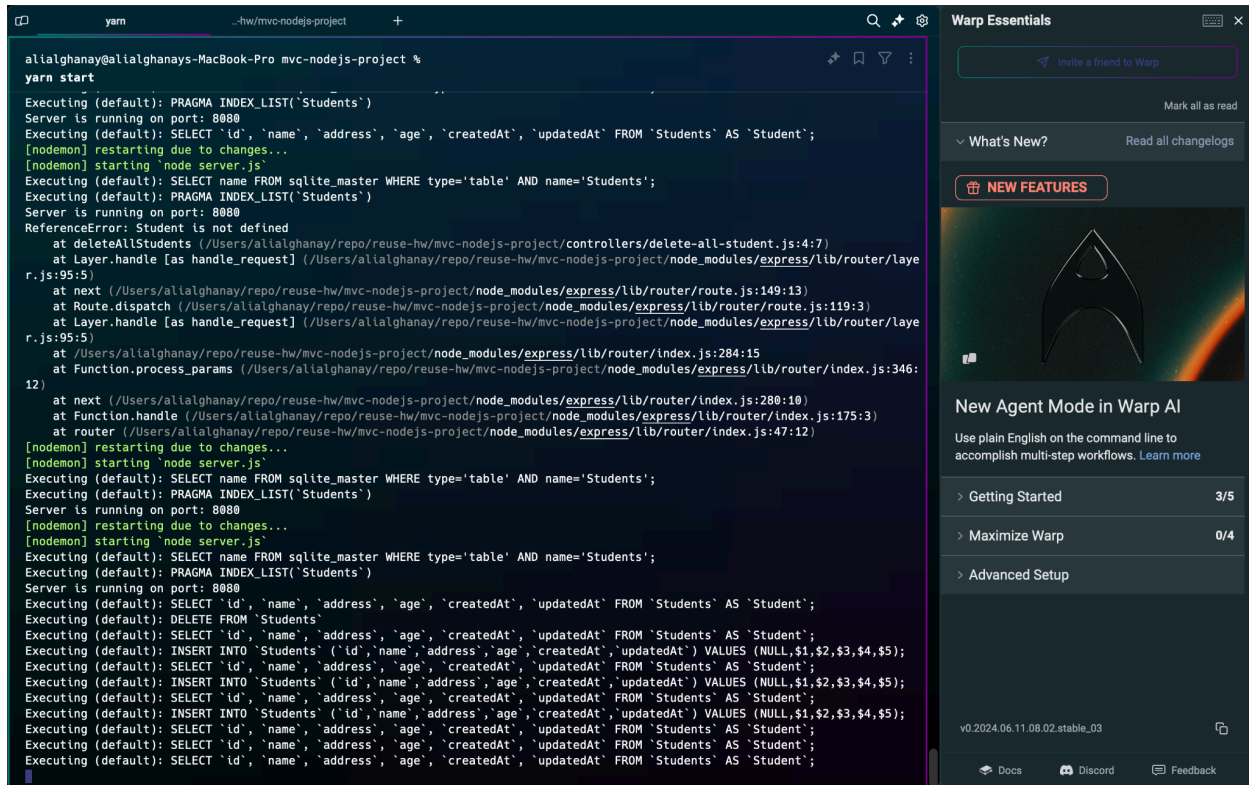


The screenshot shows a web browser window with the address bar set to 'localhost'. The browser's tab bar includes several open tabs: 'nodemon | Y...', 'ChatGPT', 'alalghanyf...', 'Node.js - D...', 'Installation |...', 'Google', 'My Drive - G...', 'مشروع الختامي...', 'anas - Googl...', and 'Reuse MVC...'. The active page is titled 'Reuse MVC Project' and features a 'Dashboard' section. At the top of the dashboard, there are two buttons: 'Add New Student' (blue) and 'Delete All Students' (red). Below these buttons, there is a list of three students, each displayed in a separate white box with a light gray border. The first student is 'علي محمد الغنائي' (Ali Mohamed Al-Ghannai), aged 23, with the address 'بئر الاسطى ميلاد' (Bir al-Ashty Milad). The second student is 'احمد مصطفى المحجوب' (Ahmed Mustafa Al-Mahjub), aged 24, with the address 'سيد خليفة' (Sidi Khalifa). The third student is 'انس فتحي المغربي' (Anas Fathi Al-Maghribi), aged 23, with the address 'شارع الزاوية' (Shar' al-Zawiya).

Student Name	Age	Address
علي محمد الغنائي	23	بئر الاسطى ميلاد
احمد مصطفى المحجوب	24	سيد خليفة
انس فتحي المغربي	23	شارع الزاوية

التاريخ: 2024-06-19.

:Terminal



التحديات والحلول

- معالجة العمليات اللازمة: تتطلب إدارة العمليات اللازمة لقاعدة البيانات القيام بعمليات متزامنة استخدام Promises و `async/await`.
- طلبات HTTP عبر العميل: كانت هناك مشكلات في البداية في استخدام `require` في المتصفح. تم حلها عن طريق تضمين Axios عبر CDN.
- تقديم النماذج: ضمان تقديم البيانات الديناميكية بشكل صحيح في العروض يتطلب فهمًا جيدًا للنحوية الخاصة بـ Pug.

الخاتمة

يبرز هذا المشروع استخدام نمط MVC في بناء تطبيق ويب بسيط لإدارة معلومات الطلاب. من خلال فصل الاهتمامات إلى نماذج وعروض ومتحكمات، يصبح قاعدة الكود أكثر تنظيمًا ونقلًا، وأكثر صيانة. كما يعزز هذا النهج إعادة استخدام البرمجيات، مما يسهل توسيع وصيانة التطبيق في المستقبل.