جافاسكربت - الوحدة السابعة: النماذج والتحقق والتخزين المحلي

النماذج هي المكان الذي يعطينا فيه الزوار معلوماتهم، والكود الجيد يعتني بكل مدخل. وتتولى أربع خطوات النموذج من النقرة الأولى إلى الزيارة التالية: الالتقاط، والتحقق، والتخزين، والاستعادة. وتجيب كل خطوة عن سؤال بسيط، وتكفي نافذة console في المتصفح لاختبار كل سطر. وفي هذه الوحدة تستخدمها مايا، مطورة منتجات لدى مزود تدريب للشركات، لبناء نموذج التسجيل في ورشة عمل.

1. الالتقاط

يرسل النموذج بياناته عندما يقدمه الزائر، وافتراضيا يرسلها المتصفح بعيدا ويحمل صفحة جديدة. والحدث submit يسمح لكودنا بأن يتولى الأمر، والدالة preventDefault في كائن الحدث توقف هذا السلوك الافتراضي ليقرر كودنا ما يحدث بعد ذلك. وتعطينا الخاصية value في كل حقل ما كتبه الزائر، وهي دائما نص string، حتى في حقل مخصص للأرقام. والسؤال هو: ماذا كتب الزائر بالضبط، ومن يتولى الأمر الآن؟ ونستمع إلى submit على النموذج نفسه وليس إلى نقرات الزر، لأن الضغط على مفتاح Enter يقدم النموذج أيضا. وقراءة القيم في كائن واحد تبقي باقي الكود مرتبا.

مثال: تسلم موظفة الاستقبال كل زائر استمارة ورقية، وعندما تعود تقرؤها قبل أن تدخل خزانة الحفظ. وتستطيع أن تتحقق من الاسم، وأن تسأل عن سطر ناقص، وأن تضع الاستمارة جانبا إذا وجدت خطأ. ولا تحفظ شيئا قبل أن تتحقق منه. ويمنح الحدث submit كودنا فرصة القراءة أولا، والدالة preventDefault هي اليد التي تمنع الاستمارة من دخول الخزانة.

في العمل: تضيف مايا مستمع submit إلى نموذج التسجيل. وتطرح سؤال الالتقاط، وتستدعي preventDefault، وتقرأ الاسم والبريد واختيار الدورة وعدد المقاعد والتعليق من الحقول إلى كائن واحد. ولأن القيم نصوص، تحول عدد المقاعد بالدالة Number كما في الوحدة الأولى من السلسلة. وصارت بيانات النموذج في يدها، جاهزة للفحص.

2. التحقق

يفحص التحقق أن البيانات مقبولة قبل استخدامها. وتقدم HTML أصلا سمات مثل required ونوع email وحد أدنى للطول، ويفحصها المتصفح قبل تقديم النموذج. ويضيف كودنا القواعد التي لا تستطيع HTML التعبير عنها، كأن يكون عدد المقاعد واحدا على الأقل ولا يزيد على ما تتسع له القاعة. والسؤال هو: ما الذي يجعل المدخل مقبولا، وكيف نقول ذلك بوضوح؟ وينبغي أن تظهر رسالة الخطأ بجانب الحقل وأن تقول ما يجب إصلاحه بكلمات بسيطة. وفحوص المتصفح تساعد الزوار وليست أمنا، لأن أي شخص يستطيع تجاوزها، ولذلك يجب أن يفحص الخادم البيانات مرة أخرى.

مثال: يتأكد ضابط الجوازات من اكتمال كل حقل ووضوحه قبل الختم، ويخبر المسافر بما ينقص بالضبط، كأن يقول إن تاريخ الميلاد غير مقروء. والرفض الغامض يرسل المسافر حائرا ويضيع وقت الجميع. والرسالة الواضحة تتيح له إصلاح أمر واحد والعودة. والأمر نفسه في النموذج، فتظهر كل رسالة بجانب الحقل وتقول ما يجب إصلاحه.

في العمل: تعتمد مايا على required ونوع email في الأساسيات. وتطرح سؤال التحقق وتضيف فحصا يجعل عدد المقاعد بين واحد وحد القاعة. وعندما يفشل فحص تكتب تحت الحقل بالخاصية textContent رسالة مثل: الرجاء إدخال عدد صحيح من المقاعد. وسيتحقق الخادم مرة أخرى، لأن فحوص المتصفح يمكن تجاوزها.

3. التخزين

يحفظ الكائن localStorage البيانات في متصفح الزائر، وتبقى فيه بعد إغلاق المتصفح. ويخزن أزواجا من مفتاح وقيمة، والقيمة دائما نص string. وتحفظ الدالة setItem، وتقرأ الدالة getItem، وتحذف الدالة removeItem مفتاحا واحدا.

حقوق الطبع والنشر محمية © - أكاديمية أساسيات (شركة أساسيات للخدمات الاستشارية)

ولتخزين كائن أو مصفوفة نحوله أولا إلى نص بالدالة JSON.stringify، ثم نعيده بالدالة JSON.parse. والسؤال هو: ما الذي يستحق الحفظ، وهل حفظه هنا آمن؟ فمثلا `localStorage.setItem("draft", JSON.stringify(draft)` يحفظ كائن المسودة نصا تحت المفتاح draft. والتخزين المحلي يخص عنوان موقع واحدا، ويستطيع أي كود على ذلك الموقع قراءته، ويستطيع الزائر تعديله، ولذلك لا مكان فيه لكلمات المرور ولا للبيانات الشخصية الحساسة.

مثال: تحفظ مفكرة على المكتب قائمة مهام الغد بعد أن تطفأ الأنوار، ويبدأ الصباح التالي من الملاحظات. وهي تحمل كلمات مكتوبة فقط، ولذلك ننسخ القائمة الطويلة أولا نصا عاديا. ويستطيع كل من يجلس إلى ذلك المكتب أن يقرأها، فلا مكان فيها لكلمة مرور خاصة. والتخزين المحلي هو هذه المفكرة داخل متصفح الزائر، والدالة JSON.stringify هي التي تنسخ.

في العمل: تحفظ مايا مسودة لاختيار الدورة والتعليق بينما يكتب الزائر. وتطرح سؤال التخزين وتحول كائن المسودة إلى نص بالدالة JSON.stringify قبل استدعاء setItem. وتترك البريد وكل ما هو حساس خارج المسودة عمدا. ويستطيع الزائر إغلاق التبويب بالخطأ دون أن يخسر عمله.

4. الاستعادة

لا يفيد الحفظ إلا إذا عادت البيانات. عند تحميل الصفحة نقرأ المفتاح بالدالة getItem، وهي تعيد null إذا لم نحفظ شيئا، فيجب أن نفحص قبل استخدام النتيجة. وإذا وجدنا نصا فإن JSON.parse يعيده كائنا، ونضع قيمه في الحقول. والسؤال هو: ماذا يجب أن يرى الزائر عندما يعود؟ وبعد تقديم ناجح نحذف المسودة بالدالة removeItem حتى لا تعود بيانات قديمة بالخطأ. وقد تجعل مسودة تالفة أو قديمة الدالة JSON.parse تفشل، والكود الحذر مستعد لهذه الحالة أيضا.

مثال: يحتفظ الفندق بتفضيلات النزيل في ملفه، وعندما يصل نزيل عائد يستقبله المكتب بغرفته المعتادة. ويلقى النزيل الجديد ترحيبا عاديا لأنه لا يوجد عنه ملف بعد، ويمسح ملف من غادر. والفندق يقرأ المعلومات ويستخدمها ويحذفها في اللحظات المناسبة. وصفحتنا تفعل الشيء نفسه: تقرأ بالدالة getItem، وتحلل بالدالة JSON.parse، وتحذف بالدالة removeItem، وتفحص null أولا.

في العمل: عند تحميل صفحة مايا تقرأ المسودة المحفوظة. وتطرح سؤال الاستعادة: ماذا يجب أن يرى الزائر عندما يعود؟ فإن وجدت مسودة تحللها وتملأ بها اختيار الدورة والتعليق، وإلا بقي النموذج فارغا. وبعد تسجيل ناجح تحذف المسودة، فيبدأ النموذج نظيفا في المرة التالية.

الخاتمة

يلتقط نموذج التسجيل عند مايا الأخطاء مبكرا ويتذكر العمل غير المكتمل. فالالتقاط يسأل ماذا كتب الزائر ومن يتولى الأمر الآن، والتحقق يسأل ما الذي يجعل المدخل مقبولا. والتخزين يسأل ما الذي يستحق الحفظ وهل حفظه هناك آمن، والاستعادة تسأل ماذا يجب أن يرى الزائر عند عودته. وفي الوحدة التالية نطلب بيانات من خادم بينما تظل الصفحة تعمل.

المصادر

- شبكة مطوري موزيلا (MDN): الخاصية Window.localStorage – https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage
- شبكة مطوري موزيلا (MDN): استخدام واجهة Web Storage – https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API/Using_the_Web_Storage_API
- شبكة مطوري موزيلا (MDN): التحقق من النماذج في جهة العميل – https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Forms/Form_validation
- شبكة مطوري موزيلا (MDN): الحدث submit في النماذج – https://developer.mozilla.org/en-US/docs/Web/API/HTMLFormElement/submit_event
- شبكة مطوري موزيلا (MDN): الدالة Storage.getItem – https://developer.mozilla.org/en-US/docs/Web/API/Storage/getItem
- شبكة مطوري موزيلا (MDN): الدالة JSON.parse – https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/parse

حقوق الطبع والنشر محمية © - أكاديمية أساسيات (شركة أساسيات للخدمات الاستشارية)