Solution by: sleekX and Aubarm (Slack)
Compiled by: Austin Tochukwu Asoluka

Grow

With Google | Andela | Pluralsight

(Your path to become a Mobile Web Specialist)

Mobile Web Specialist Training

ALC 4.0: Phase 1 Challenge Solution - Mobile Web

Description: This challenge is aimed at having you roll up your sleeves and do some real coding practice based on what you've learnt from the contents available to you on the Pluralsight platform. The hands-on project will gauge your ability to apply the knowledge you've acquired and also challenge you in this learning journey.

Solution by: sleekX and Aubarm (Slack)
Compiled by: Austin Tochukwu Asoluka

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <meta http-equiv="X-UA-Compatible" content="ie=edge" />
        <title>Mini App</title>
        <style>
                body {
                        margin: 0;
                        padding: 1em;
                        background-color: #fff;
                }

                div[data-cart-info] span {
                        display: inline-block;
                        vertical-align: middle;
                }

                span.material-icons  {
                        font-size: 150px;
                }

                div[data-credit-card] {
                        width: 435px;
                        min-height: 240px;
                        border-radius: 10px;
                        background-color: #5d6874;
                }
```

```
img[data-card-type] {
        display: block;
        width: 120px;
        height: 60px;
}

div[data-cc-digits] {
        margin-top: 2em;
}

div[data-cc-digits] input {
        color: #fff;
        font-size: 2em;
        line-height: 2em;
        border: none;
        background: none;
        margin-right: 0.5em;
}

div[data-cc-info]{
        margin-top: 1em;
}

div[data-cc-info] input {
        color: #fff;
        font-size: 1.2em;
        border: none;
        background: none;
}
```

```
div[data-cc-info] input:nth-child(2){
        padding-right: 10px;
        float: right;
}

button[data-pay-btn]{
        position: fixed;
        width: 90%;
        border: 1px solid;
        bottom: 20px;
}

[data-cart-info],
[data-credit-card] {
        transform: scale(0.78);
        margin-left: -3.4em;
}

[data-cc-info] input:focus,
[data-cc-digits] input:focus {
        outline: none;
}


.mdc-card__primary-action,
.mdc-card__primary-action:hover {
        cursor: auto;
        padding: 20px;
        min-height: inherit;
```

```
            }

            [data-credit-card] [data-card-type] {
                transition: width 1.5s;
                margin-left: calc(100% - 130px);
            }

            [data-credit-card].is-visa {
                background: linear-gradient(135deg, #622774 0%, #c53364 100%);
            }

            [data-credit-card].is-mastercard {
                background: linear-gradient(135deg, #65799b 0%, #5e2563 100%);
            }

            .is-visa [data-card-type],
            .is-mastercard [data-card-type] {
                width: auto;
            }

            input.is-invalid,
            .is-invalid input {
                text-decoration: line-through;
            }

            ::placeholder {
                color: #fff;
            }
        </style>
</head>
```

```
<body>
        <div data-cart-info>
                <h1 class="mdc-typography--headline4">
                        <span class="material-icons">shopping_cart</span>
                        <span data-bill></span>
                </h1>
        </div>

        <div data-credit-card class="mdc-card mdc-card--outlined">
                <div class="mdc-card__primary-action">
                        <img data-card-type src="http://placehold.it/120x60.png?ext=Card" alt="">
                        <div data-cc-digits>
                                <input type="text" size="4" placeholder="----">
                                <input type="text" size="4" placeholder="----">
                                <input type="text" size="4" placeholder="----">
                                <input type="text" size="4" placeholder="----">
                        </div>

                        <div data-cc-info>
                                <input type="text" size="20" placeholder="Name Surname">
                                <input type="text" size="6" placeholder="MM/YY">
                        </div>

                </div>
        </div>

        <button class="mdc-button" data-pay-btn>Pay & Checkout Now</button>
```

```
<script>
        const supportedCards = {
                visa, mastercard
        };


        const countries = [
        {
                code: "US",
                currency: "USD",
                country: 'United States'
        },
        {
                code: "NG",
                currency: "NGN",
                country: 'Nigeria'
        },
        {
                code: 'KE',
                currency: 'KES',
                country: 'Kenya'
        },
        {
                code: 'UG',
                currency: 'UGX',
                country: 'Uganda'
        },
        {
                code: 'RW',
```

```
        currency: 'RWF',
        country: 'Rwanda'
},
{
        code: 'TZ',
        currency: 'TZS',
        country: 'Tanzania'
},
{
        code: 'ZA',
        currency: 'ZAR',
        country: 'South Africa'
},
{
        code: 'CM',
        currency: 'XAF',
        country: 'Cameroon'
},
{
        code: 'GH',
        currency: 'GHS',
        country: 'Ghana'
}
];

const appState = {};

const formatAsMoney = (amount, buyerCountry) =>
{
        const country = countries.find(
```

```
        country =>country.country == buyerCountry);
        if(country){
                return amount.toLocaleString('en-'+country.code, {style: 'currency', currency: country.currency});
        } else {
                return amount.toLocaleString('US', {style:'currency', currency: 'USD'});
        }
}

const flagIfInvalid = (field, isValid) => {
        isValid ?
        field.classList.remove('is-invalid') : field.classList.add('is-invalid');
}

const expiryDateFormatIsValid = (target) => {
        if(target.value){
                if((target.value.indexOf('/') == 2) && (target.value.length == 5)){
                        // care to check if the first 2 and the last 2 chars are numbers?
                        return true;
                }
                return false;
        }
}
```

```
const isFuturisticDate = (date) => {
	const currDate = new Date();
	const inputMonth = parseInt(date.value.split('/')[0], 10);
	if (inputMonth > 12 || inputMonth < 1) {
		return false
	}
	const inputYear = parseInt(date.value.split('/')[1], 10);
	const twoDigitFullYear = parseInt(currDate.getFullYear().toString().substr(-2), 10);
	const twoDigitMonth = currDate.getMonth() + 1;
	return (inputYear > twoDigitFullYear )|| (inputYear == twoDigitFullYear && inputMonth >
	twoDigitMonth);
}

const detectCardType = ({target}) => {
	const card = document.querySelector('div[data-credit-card]');
	const cardLogo = document.querySelector('[data-card-type]');
	if (target.value.startsWith('4')){
		card.classList.remove('is-mastercard');
		card.classList.add('is-visa');
		cardLogo.src = supportedCards.visa;
		return 'is-visa';
	} else if (target.value.startsWith('5')) {
		card.classList.remove('is-visa');
		card.classList.add('is-mastercard');
		cardLogo.src = supportedCards.mastercard;
		return 'is-mastercard';
	}
}
```

```
const validateCardExpiryDate = ({target}) => {
	if(target.value){
		const isValidExpiryDate = expiryDateFormatIsValid(target) && isFuturisticDate(target);
		flagIfInvalid(target, isValidExpiryDate);
		return isValidExpiryDate;
	}
}


const validateCardHolderName = ({target}) => {
	let fullName = target.value;
	console.log(fullName);
	if(fullName && fullName.match(/^[A-Z][a-z]+\s[A-Z][a-z]+$/)){
		console.log('is-true')
		flagIfInvalid(target, true);
		return true;
	}

	flagIfInvalid(target, false)
	return false;
}
```

Solution by: sleekX and Aubarm (Slack)
Compiled by: Austin Tochukwu Asoluka

```
const validateWithLuhn = digits => {
	let isOtherDigit = false;

	if (digits.length != 16) {
		return false;
	}

	if(!(/^\d+$/.test(digits.join('')))) {
		return false;
	}

	const stepOneArr = [];
	for(i = digits.length - 1; i >= 0; i--){
		if(isOtherDigit == false){
			stepOneArr.unshift(digits[i]);
		} else {
			let newOtherDigit = digits[i] * 2;
			if (newOtherDigit > 9){
				newOtherDigit = newOtherDigit - 9;
			}
		stepOneArr.unshift(newOtherDigit);
		}
	isOtherDigit = !isOtherDigit;
	}

	const sum = stepOneArr.reduce((accumulator, currentVal) => {return accumulator + currentVal}, 0);

	if (sum % 10 == 0){
		return true;  // true means valid
```

```
        }
        return false;  // false means.. well, you know
}

const validateCardNumber = () => {
        const digitsElArr = Array.from(document.querySelectorAll('[data-cc-digits] input'));
        const digitsStr = digitsElArr.reduce((values, currentEl) => {
                return values + currentEl.value
        }, '');
        const digits = digitsStr.split('').map(digit => parseInt(digit, 10));
        const isValidNumber = validateWithLuhn(digits);
        const digitInputsDiv = document.querySelector('[data-cc-digits]');
        if (isValidNumber) {
                digitInputsDiv.classList.remove('is-invalid');
        } else {
                digitInputsDiv.classList.add('is-invalid');
        }
        return isValidNumber;
}

const uiCanInteract = () => {
        document.querySelectorAll('[data-cc-digits] input')[0].addEventListener('blur', detectCardType);
        document.querySelectorAll('[data-cc-info] input')[0].addEventListener('blur', validateCardHolderName);
        document.querySelectorAll('[data-cc-info] input')[1].addEventListener('blur', validateCardExpiryDate);
        document.querySelector('[data-pay-btn]').addEventListener('click', validateCardNumber);
        document.querySelectorAll('[data-cc-digits] input')[0].focus();
}
```

```
            const displayCartTotal = ({results}) => {
                const [data,]= results;
                const {itemsInCart, buyerCountry} = data;
                appState.items = itemsInCart;
                appState.country = buyerCountry;
                appState.bill = itemsInCart.reduce((accum, currVal) => {return accum + (currVal.qty * currVal.price )}, 0);
                appState.billFormatted = formatAsMoney(appState.bill, appState.country);
                document.querySelector('[data-bill]').textContent = appState.billFormatted;
                uiCanInteract();
            }

            const fetchBill = () => {
                const api = "https://randomapi.com/api/006b08a801d82d0c9824dcfdfdfa3b3c";
                fetch(api).then(resp => resp.json()).then(displayCartTotal).catch(err => console.log(err))
            }

            const startApp = () => {
                fetchBill();
            };

            startApp();
        </script>
</body>
</html>
```