

Registration Form

First Name

Last Name

Email

Cell

Street Address

City

Area

ZIP

Passwrod

Confirm Password

☐ I agree to the [Terms of Services](#) ☐ Signup for Newsletter

Register

Registration Form

First Name

Enter First Name

* First Name Field Must Not Be Empty

Last Name

Enter Last Name

* Last Name Field Must Not Be Empty

Email

Enter Email

* Email Field Must Not Be Empty

Cell

Enter Cell No

* Cell Phone Field Must Not Be Empty

Street Address

Street Address

* Street Address Field Must Not Be Empty

City

Dhaka



Area

Mirpur



ZIP

Passwrod

* Password Field Must Not Be Empty

Confirm Password

* Confirm Password Field Must Not Be Empty

☐ I agree to the [Terms of Services](#)

☐ Signup for Newsletter

Register

Registration Form

First Name

Ahid12

* Invalid First Name Format

Last Name

Rad`-

* Invalid Last Name Format

Email

asdf@asd

* Invalid Email Format

Cell

jkl

* Invalid Cell Phone Format

Street Address

!@#\$

* Invalid Street Address Format

City

Dhaka

Area

Mirpur

ZIP

Passwrod

....

* Passwords does not match

Confirm Password

.....

* Passwords does not match

☐

I agree to the [Terms of Services](#)

☐

Signup for Newsletter

```
const regForm = document.querySelector('#reg-form');
let allClear = false;

const emailPattern = /^\\w+@[a-zA-Z_]+?\\. [a-zA-Z]{2,3}$/;
const namePattern = /^[a-zA-Z.]{2,30}$/;
const cellPattern = /^[0-9-+ ]{11,20}$/;
const emailPattern2 = /^((([^<>()\\]{'^?\\\\. , !|//#%*-+=&;:\\\\s@\\"]+(\\. [^<>()\\]\\\\\\\\. , ;:\\\\s@\\"]+)*)| (\\\".+\\\\ [a-z0-9](?:[a-z0-9-]*[a-z0-9]))?)?/
const addressPattern = /^[a-zA-Z0-9-.,()# ]{2,30}$/;

const firstName = regForm.elements.firstname;
let firstNameErr = document.querySelector('#firstNameErr');

firstName.addEventListener('input', function () {
    validateInput(firstName.value, namePattern, firstNameErr, 'First Name');
})

firstName.addEventListener('focusout', function () {
    validateIsEmpty(firstName.value, firstNameErr, 'First Name');
})
```

```
function validateInput(input, regxPattern, errSpan, errMsg) {
    if (!input) {
        errSpan.classList.remove("hidden");
        errSpan.innerHTML = `* ${errMsg} Field Must Not Be Empty`;
        allClear = false;
    } else if (!regxPattern.test(input)) {
        errSpan.classList.remove("hidden");
        errSpan.innerHTML = `* Invalid ${errMsg} Format`;
        allClear = false;
    } else {
        errSpan.classList.add("hidden");
        allClear = true;
    }
}
```

```
function validateIsEmpty(input, errSpan, errMsg) {
    if (!input) {
        errSpan.classList.remove("hidden");
        errSpan.innerHTML = `* ${errMsg} Field Must Not Be Empty`;
        allClear = false;
    }
}
```

```
function validatePasswordMatch() {
    if (password.value !== confirmPassword.value) {
        passwordErr.innerHTML = confirmPasswordErr.innerHTML = "* Passwords does not match";
        passwordErr.classList.remove("hidden");
        confirmPasswordErr.classList.remove("hidden");
    } else {
        passwordErr.classList.add("hidden");
        confirmPasswordErr.classList.add("hidden");
    }
}
```

```
const confirmPassword = regForm.elements.confirmPassword;
let confirmPassErr = document.querySelector('#confirmPassErr');

confirmPassword.addEventListener('focusout', function () {
    validateIsEmpty(confirmPassword.value, confirmPassErr, 'Confirm Password');
    validatePasswordMatch();
})

confirmPassword.addEventListener('input', function () {
    validatePasswordMatch();
})
```