Medium password

8+ chars, at least one letter and one digit.

```
^(?=.*[A-Za-z])(?=.*\d).{8,}$
```

Token-by-token breakdown

Token	Meaning
`^`	start of string (or line in multiline mode)
`(?=`	start of a positive lookahead — followed by
`.*`	zero or more of any character
`[A-Za-z]`	any of: uppercase letters, lowercase letters
`)`	end of group
`(?=`	start of a positive lookahead — followed by
`.*`	zero or more of any character
`\d`	any digit (0–9)
`)`	end of group
`.{8,}`	8 or more times of any character
`$`	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript

```
const re = /^(?=.*[A-Za-z])(?=.*\d).{8,}$/;
re.test(value);
```

Python

```
import re
re.match(r"^(?=.*[A-Za-z])(?=.*\d).{8,}$", value)
```

Java

```
Pattern.compile("^(?=.*[A-Za-z])(?=.*\\d).{8,}$").matcher(value).matches();
```

C# / .NET

```
Regex.IsMatch(value, @"^(?=.*[A-Za-z])(?=.*\d).{8,}$");
```

Go

```
regexp.MustCompile(`^(?=.*[A-Za-z])(?=.*\d).{8,}$`).MatchString(value)
```

Ruby

```
/^(?=.*[A-Za-z])(?=.*\d).{8,}$/.match?(value)
```

PHP

```
preg_match('~^(?=.*[A-Za-z])(?=.*\d).{8,}$~', $value);
```

Common pitfalls

- **Anchored to the whole string.** This pattern uses ^ and $, so it requires the entire input to match. To find it inside a longer text, drop the anchors and use the global (g) flag.
- **ASCII letters only.** [a-zA-Z] excludes accented and non-Latin letters (é, ü, ß, ñ, and non-Latin scripts). For international input use Unicode properties like \p{L} with the u flag.
- **Greedy ".*".** A greedy .* / .+ can match more than intended. Use a lazy version (.*?) or a negated class ([^…]) to stop at the right place.
- **Escape it correctly per language.** In Java and JavaScript strings each backslash must be doubled (\\d); in Python, Go, and C# use raw/verbatim strings so the backslashes survive.

• **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the medium password against a source of truth (database, API, or checksum) where it matters.