REGEX CHEAT SHEET

regexguide.com/patterns/india-speedpost.html

Speed Post tracking

India Post Speed Post — starts with EE/EM and ends IN.

```
^E[EM]\d{9}IN$
```

Token-by-token breakdown

Token	Meaning
`^`	start of string (or line in multiline mode)
`E`	literal text “E”
`[EM]`	any of: “E”, “M”
`\d{9}`	exactly 9 times: digit (0–9)
`IN`	literal text “IN”
`$`	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript

```
const re = /^E[EM]\d{9}IN$/;
re.test(value);
```

Python

```
import re
re.match(r"^E[EM]\d{9}IN$", value)
```

Java

```
Pattern.compile("^E[EM]\\d{9}IN$").matcher(value).matches();
```

C# / .NET

```
Regex.IsMatch(value, @"^E[EM]\d{9}IN$");
```

Go

```
regexp.MustCompile(`^E[EM]\d{9}IN$`).MatchString(value)
```

Ruby

```
/^E[EM]\d{9}IN$/.match?(value)
```

PHP

```
preg_match('~^E[EM]\d{9}IN$~', $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.
- **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 speed post tracking against a source of truth (database, API, or checksum) where it matters.