

**Sri Lanka Institute of Information  
Technology**

**Programing Paradigms  
SE2062**

**EBNF of DSL and HTML Form Generator  
with Lex and Yacc**

**Manula Cooray  
IT23194830**

program ::= "form" IDENTIFIER "{" form\_body validate\_block?"}"

form\_body ::= meta\_section? section\_list

meta\_section ::= { "meta" IDENTIFIER "=" STRING ";" }

section\_list ::= { "section" IDENTIFIER "{" field\_list "}" }

field\_list ::= { field }

field ::= "field" IDENTIFIER ":" field\_type attribute\_block?  
";"

          | "field" IDENTIFIER ":" field\_type string\_list  
attribute\_block? ";"

field\_type ::= "text" | "password" | "number" | "email" | "date"  
              | "checkbox" | "radio" | "dropdown" | "textarea" |  
"file"

string\_list ::= "[" STRING { "," STRING } "]"

attribute\_block ::= { attribute }

attribute ::= "required"  
              | "min" "=" (NUMBER | STRING)  
              | "max" "=" (NUMBER | STRING)  
              | "default" "=" (STRING | BOOL)  
              | IDENTIFIER "=" (NUMBER | STRING)

`validate_block ::= "validate" "{" { validation_rule } "}"`

`validation_rule ::= "if" condition "{" "error" STRING ";" "}"`

`condition ::= IDENTIFIER comparator NUMBER`

`comparator ::= "<" | ">" | "=="`

`STRING ::= "'" (any character except quote)+ "'"`

`IDENTIFIER ::= (letter)(letter | digit | "_")*`

`NUMBER ::= digit+`

`BOOL ::= "true" | "false"`

### ❖ Program Structure

```
program ::= "form" IDENTIFIER "{" form_body validate_block? "}"
```

- Defines the entire form.
- Starts with form, a name (IDENTIFIER), followed by the form body.
- Optionally ends with a validation block.

### ❖ Form Body

```
form_body ::= meta_section? section_list
```

- Contains optional metadata and one or more sections.

### ❖ Metadata

```
meta_section ::= { "meta" IDENTIFIER "=" STRING ";" }
```

- Allows adding form-level metadata like author or version.

### ❖ Sections

```
section_list ::= { "section" IDENTIFIER "{" field_list "}" }
```

- Defines named sections within the form.
- Each section contains a list of fields.

### ❖ Fields

```
field_list ::= { field }
```

```
field ::= "field" IDENTIFIER ":" field_type attribute_block? ";"  
        | "field" IDENTIFIER ":" field_type string_list  
        attribute_block? ";"
```

- Fields are form inputs (text, number, etc.).
- Some field types (like dropdown, radio) include option lists.

❖

### ❖ Field Types

```
field_type ::= "text" | "password" | "number" | ...
```

- Specifies the type of input (e.g., email, checkbox, file, etc.).

### ❖ Option Lists

```
string_list ::= "[" STRING { "," STRING } "]"
```

- Used in fields like radio and dropdown.
- Specifies selectable options.

### ❖ Attributes

```
attribute_block ::= { attribute }
```

```
attribute ::= "required" | "min=..." | "default=..." | ...
```

- Additional settings for fields like required, min, max, etc.

### ❖ Validation Rules

```
validate_block ::= "validate" "{" { validation_rule } "}"
```

- Optional block to define logic checks on input.

### ❖ Conditions

```
validation_rule ::= "if" condition "{" "error" STRING ";" "}"
```

```
condition ::= IDENTIFIER comparator NUMBER
```

```
comparator ::= "<" | ">" | "=="
```

- Simple validation checks like `if age < 18`, showing an error if `true`.

### ❖ Terminals

```
STRING ::= "\"" text "\""
```

```
IDENTIFIER ::= letter followed by letters/digits/underscores
```

```
NUMBER ::= digits
```

```
BOOL ::= "true" | "false"
```

- These are the basic units of the language: strings, IDs, numbers, booleans.

## Terminals

Terminal Type	Examples
Keywords	form, meta, section, field, validate, if, error, required, true, false
Symbols	{, }, [, ], :, ;, =, <, >, ==, ,
Tokens	STRING, IDENTIFIER, NUMBER, BOOL

## Non-terminals

program	field_type
form_body	string_list
meta_section	attribute_block
section_list	attribute
field_list	validate_block
field	validation_rule
	condition
	comparator