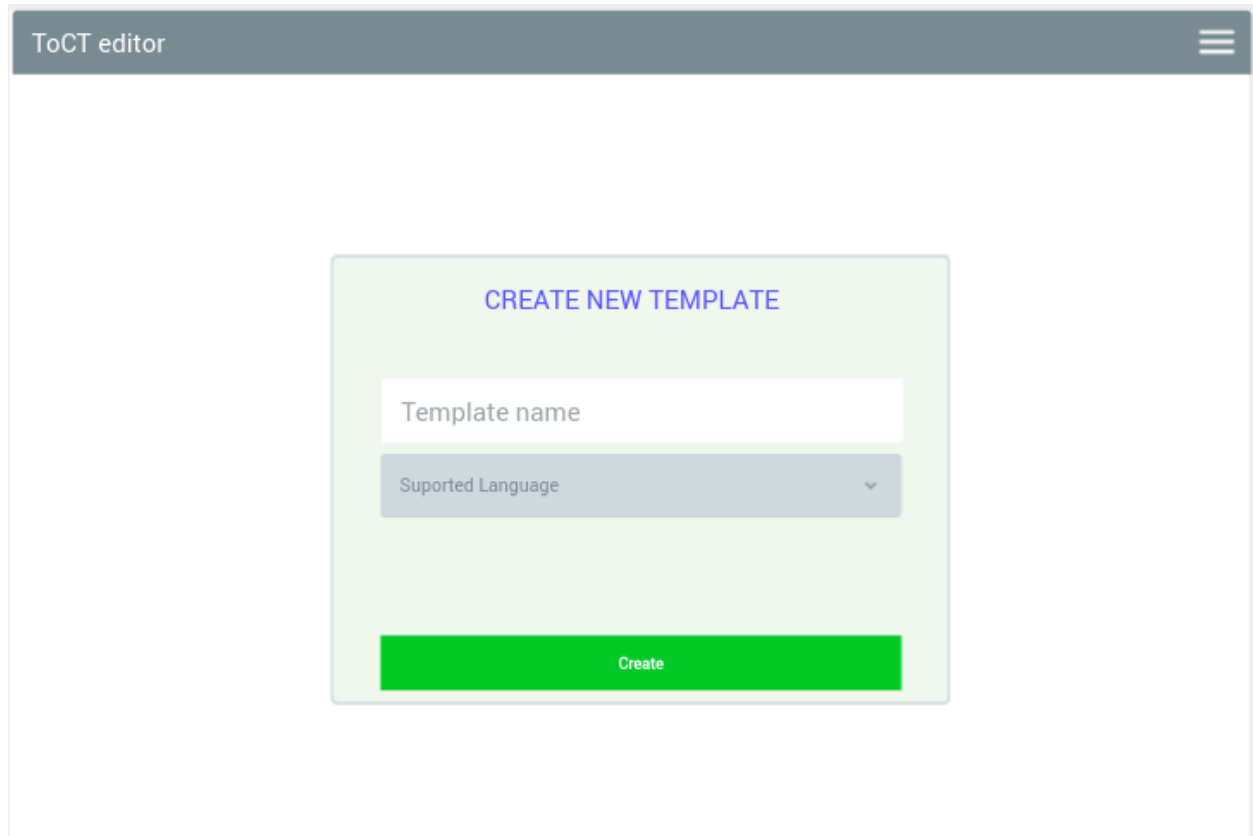


Link to Interactive prototype

<https://www.fluidui.com/editor/live/preview/cF85dkIYM2dDM3ZEOWZPUzFWY1VKMjlQT0VEUkNCQ21Tbg==>

Screen 1: Home



The screenshot shows a web application titled "ToCT editor" in a dark grey header bar. On the right side of the header is a hamburger menu icon. The main content area is white and contains a light green rounded rectangle centered on the page. Inside this rectangle, the text "CREATE NEW TEMPLATE" is displayed in blue at the top. Below this text is a form with three elements: a white text input field with the placeholder text "Template name", a grey dropdown menu with the text "Suported Language" and a downward arrow, and a bright green button at the bottom with the text "Create".

Interactive components:

1. Textfield-*Template name*
2. Dropdownbox-*Supported language*
3. Button-*Create*

Screen 2: Choose part

ToCT editor

SELECT TEMPLATE PART TO CREATE

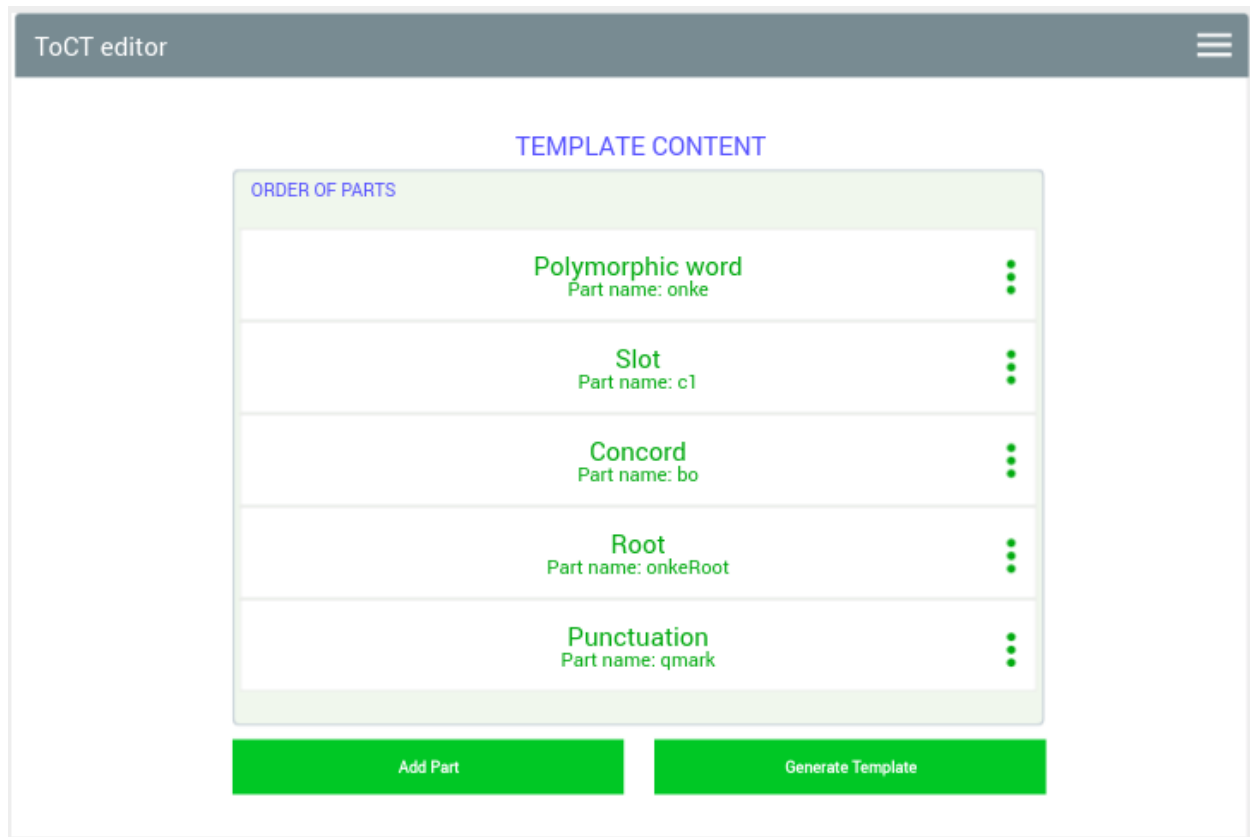
Slot	Unimorphic Word	Punctuation
Root	Polymorphic Word	Concord
Phrase	Copula	Unimorphic Affix

Cancel

Interactive components:

1. Button-*Slot*
2. Button-*Unimorphic word*
3. Button-*Punctuation*
4. Button-*Root*
5. Button-*Polymorphic word*
6. Button-*Concord*
7. Button-*Phrase*
8. Button-*Copula*
9. Button-*Unimorphic-Affix*
10. Button-*Cancel*

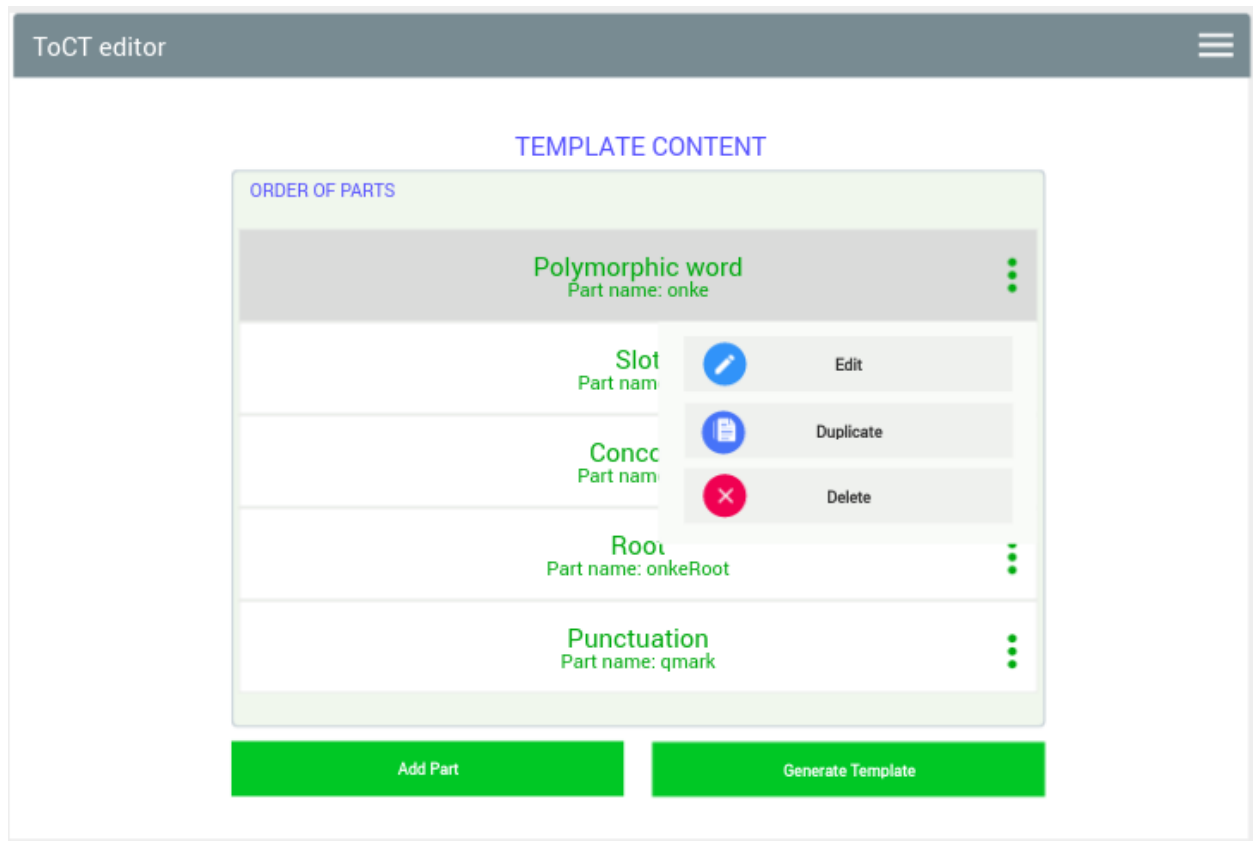
Screen 3: Template content



Interactive components:

1. Button-*Add Part*
2. Button-*Generate Template*
3. Movable part-boxes

Screen 3: Template content - Kebab menu on click



Screen 3: Template content - Add part

ToCT editor

TEMPLATE CONTENT

ORDER OF PARTS

Polymorphic word
Part name: onke

Slot
Part name: c1

Polymorphic word

Slot

Concord

More Options . . .

oot

rk

Add Part

Generate Template

Design Flow

Create template:

Button-Create

Layout flow

Screen 1: Home -> Screen 2: Choose Part

(1)Create part (Slot):

Button-*Slot* -> Button-Add

Layout flow

Screen 2: Choose Part -> Screen-Slot-Properties -> Screen 2: Choose Part

ToCT editor

Slot

Unimorphic Word

Punctuation

Root

Polymorphic Word

Concord

Phrase

Copula

Unimorphic Affix

Part Properties

Show turtle syntax

Part name

Has label

Has next part

```
< c1 > a toct:Slot
; toct:hasLabel "C1"^^xsd:string
; toct:hasNextPart < c2 > .
```

Back

Add

(2) Create part (Unimorphic word):

Button-*Unimorphic word* -> Button-Add

Layout flow

Screen 2: Choose Part -> Screen-Unimorphic-Word-Properties -> Screen 2:

Choose Part

ToCT editor

Slot

Unimorphic Word

Punctuation

Root

Polymorphic Word

Concord

Phrase

Copula

Unimorphic Affix

Part Properties

Show turtle syntax ☒

Part name

Has value

Has next part

```
< ngabe > a toct:UnimorphicWord
; toct:hasValue "Ingabe"^^xsd:string
; toct:hasNextPart < onke > .
```

Back

Add

(3) Create part (Punctuation):

Button-*Punctuation word* -> Button-Add

Layout flow

Screen 2: Choose Part -> Screen-Punctuation-Properties -> Screen 2: Choose

Part

ToCT editor

Slot

Unimorphic Word

Punctuation

Root

Polymorphic Word

Concord

Phrase

Copula

Unimorphic Affix

Part Properties

Show turtle syntax ☒

Part name

Has value

```
< qmark > a toct:Punctuation
; toct:hasValue "?"^^xsd:string .
```

Back

Add

(4) Create part (Root):

Button-Root -> Button-Add

Layout flow

Screen 2: Choose Part -> Screen-Root-Properties -> Screen 2: Choose Part

ToCT editor

Slot

Unimorphic Word

Punctuation

Root

Polymorphic Word

Concord

Phrase

Copula

Unimorphic Affix

Part Properties

Show turtle syntax ☒

Part name

Has value

```
< qmark > a toct:Punctuation  
; toct:hasValue "?"^^xsd:string .
```

Back

Add

(5) Create part (Polymorphic Word):

Button-*Polymorphic-Word* -> Button-Add

Layout flow

Screen 2: Choose Part -> Screen-Polymorphic-Word-Properties -> Screen 2:

Choose Part

ToCT editor

Slot

Unimorphic Word

Punctuation

Root

Polymorphic Word

Concord

Phrase

Copula

Unimorphic Affix

Part Properties

Show turtle syntax ☒

Part name

Relies on

Has first part

Has last part

Has next part

```
< onke > a toct:PolymorphicWord
; toct:reliesOn < c1 >
; toct:hasFirstPart < bo >
; toct:hasLastPart < onkeRoot >
; toct:hasNextPart < c1 > .
```

Back

Add

(6) Create part (Concord):

Button-*Concord* -> Button-Add

Layout flow

Screen 2: Choose Part -> Screen-Concord-Properties -> Screen 2: Choose Part

ToCT editor

Slot

Unimorphic Word

Punctuation

Root

Polymorphic Word

Concord

Phrase

Copula

Unimorphic Affix

Part Properties

Show turtle syntax ☒

Part name

Has label

Has concord type

Has next part

```
< c2Sub > a toct:Concord
; toct:hasLabel 'subjC'^xsd:string
; cao:hasConcordType < subjCConType >
; toct:hasNextPart < c2Slot > .
```

Back

Add

(6) Create part (Phrase):

Button-*Phrase* -> Button-Add

Layout flow

Screen 2: Choose Part -> Screen-Phrase-Properties -> Screen 2: Choose Part

ToCT editor

Slot

Unimorphic Word

Punctuation

Root

Polymorphic Word

Concord

Phrase

Copula

Unimorphic Affix

Part Properties

Show turtle syntax

Part name

Has value

Has next part

```
< ingabep > a toct:Phrase
; toct:hasValue "Ingabe noma"^^xsd:string
; toct:hasNextPart < yiphi > .
```

Back

Add

(6) Create part (Copula):

Button-*Copula* -> Button-Add

Layout flow

Screen 2: Choose Part -> Screen-Copula-Properties -> Screen 2: Choose Part

ToCT editor

Slot

Unimorphic Word

Punctuation

Root

Polymorphic Word

Concord

Phrase

Copula

Unimorphic Affix

Part Properties

Show turtle syntax

Part name

Has label

Has next part

```
< cop > a toct:Copula
; toct:hasLabel "COP"^^xsd:string
; toct:hasNextPart < c1 > .
```

Back

Add

(6) Create part (Unimorphic Affix):

Button-*Unimorphic-Affix* -> Button-Add

Layout flow

Screen 2: Choose Part -> Screen-Unimorphic-Affix-Properties -> Screen 2:
Choose Part

ToCT editor

Slot

Unimorphic Word

Punctuation

Root

Polymorphic Word

Concord

Phrase

Copula

Unimorphic Affix

Part Properties

Show turtle syntax

Part name

Has value

```
< phi > a toct:UnimorphicAffix
; toct:hasValue "phi"^^xsd:string .
```

Back

Add