

دانشکده مهندسی کامپیوتر

**طراحی و پیاده سازی رابط گرافیکی کاربر برای نرم افزار کلیپس با ویژگی تطبیق پذیر بودن**

پروژه پایانی برای دریافت درجه کارشناسی در رشته مهندسی کامپیوتر

علی اسدی

استاد راهنما

دکتر محمد رضا کنگاوری

پاییز1402

V

تأییدیه ی هیأت داوران جلسه ی دفاع از پایان نامه

نام دانشکده: دانشکده مهندسی کامپیوتر

نام دانشجو: علی اسدی

عنوان پایان نامه: طراحی و پیاده سازی رابط گرافیکی کاربر برای نرم افزار کلیپس با ویژگی تطبیق پذیر بودن

تاریخ دفاع: پاییز ۱۴۰۲

رشته: مهندسی کامپیوتر

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ردیف | سمت | نام و نام خانوادگی | مرتبه دانشگاهی | دانشگاه | امضا |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |

تأییدیه ی صحت و اصالت نتایج

باسمه تعالی

اینجانب علی اسدی به شماره دانشجویی 96521038 دانشجوی رشته مهندسی کامپیوتر مقطع تحصیلیکارشناسی تأیید مینمایم كه كلیه ی نتایج این پایان نامه حاصل كار اینجانب و بدون هرگونه دخل و تصرف است و موارد نسخه برداری شده از آثار دیگران را با ذكر كامل مشخصات منبع ذكر كرده ام. درصورت اثبات خلاف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاكم ( قانون حمایت از حقوق مؤلفان و مصنفان و قانون ترجمه و تكثیر كتب و نشریات و آثار صوتی،ضوابط و مقررات آموزشی، پژوهشی و انضباطی ... ) با اینجانب رفتار خواهد شد و حق هرگونه اعتراض درخصوص احقاق حقوق مکتسب و تشخیص و تعیین تخلف و مجازات را از خویش سلب مینمایم. در ضمن، مسؤولیت هرگونه پاسخگویی به اشخاص اعم از حقیقی و حقوقی و مراجع ذی صلاح (اعم از اداری و قضایی) به عهده ی اینجانب خواهد بود و دانشگاه هیچ گونه مسؤولیتی در این خصوص نخواهد داشت

نام و نام خانوادگی: علی اسدی

تاریخ و امضا

مجوز بهره برداری از پایان نامه

بهره برداری از این پایان نامه در چهارچوب مقررات كتابخانه و با توجه به محدودیتی كه توسط استاد راهنما

به شرح زیر تعیین میشود، بلامانع است:

□ بهره برداری از این پایان نامه برای همگان بلامانع است.

□ بهره برداری از این پایان نامه با اخذ مجوز از استاد راهنما، بلامانع است.

□ بهره برداری از این پایان نامه تا تاریخ .................................... ممنوع است.

استاد راهنما

تاریخ:

امضا:

تقدیم به

پدر و مادرم

تشکر و قدردانی

سپاس خداوندگار حکیم را که با لطف بی کران خود، آدمی را زیور عقل آراست

. در آغاز وظیفه خود می دانم از زحمات بی دریغ استاد راهنمای خود، جناب آقای دکتر کنگاوری صمیمانه تشکر و قدردانی کنم که قطعاً بدون راهنمایی های ارزنده ایشان، این مجموعه به انجام نمیرسید

در پایان، بوسه میزنم بر دستان خداوندگاران مهر و مهربانی پدر و مادر عزیزم و بعد از خدا، ستایش میکنم وجود مقدس شان را و تشکر می کنم از خانواده عزیزم به پاس عاطفه سرشار و گرمای امیدبخش وجودشان، که بهترین پشتیبان من بودند.

علی اسدی

پاییز 1402

چکیده

کلیپس CLIPS یک محیط نرم افزاری برای تولید سیستم های خبره است که فاقد یک رابط گرافیکی بر خط است. کلپیس یک زبان برنامه نویسی توصیفی است که برخلاف زبان های رویه ای که در آن ها باید روند و چگونگی کار برای سیستم بیان شود تا سیستم ساخته شود، در این زبانها تعریف صورت مسئله درسیستم، منجر به حل مسئله می گردد

هدف این پروژه طراحی و تولید یک رابط کاربری گرافیکی برای کلیپس است که استفاده از کلیپس را برای طراحان و کاربران تسهیل می نماید.

برای تولید این رابط کاربری از زبان برنامه نویسی جاوا اسکریپت و کتابخانه react استفاده شده و در نهایت خروجی دو فایل به زبان های پایتون (.py) و کلیپس (.clp) بوده که کاربران می توانند از آن ها در محیط پایتون و برنامه کلیپس استفاده کنند.

برای پیاده سازی یک برنامه خبره نیاز به یک سری اطلاعات داریم که چگونگی پیاده سازی این سیستم را برای ما تشریح کند. این اطلاعات توسط تحقیق و پژوهش از فرد خبره در آن زمینه به دست می آید که قوانین ما را تشکیل می دهند

پس از آن که این اطلاعات را کسب کردیم نیاز به دسته بندی و ایجاد یک درخت تصمیم برای سیستم خبره خود داریم. این درخت می تواند همان سناریو سیستم شما باشد که نحوه کسب دانش و واکنش ها به دانش ها را مشخص می کند.

فهرست مطالب

1. مقدمه..........................................................................................................................................................................11
2. برنامه کلیپس............................................................................................................................................................13
3. راه اندازی برنامه.......................................................................................................................................................30
4. نحوه کار برنامه..........................................................................................................................................................32
5. پیاده سازی یک سناریو...........................................................................................................................................36
6. منابع و مراجع...........................................................................................................................................................52

۱..مقدمه

سیستم خبره یک برنامه کامپیوتری هوشمند است که از دانش و روش های استنتاج برای حل مسائلی استفاده می کند که به دلیل مشکل بودن نیاز به تجربه و مهارت انسان دارد.این سیستم یک واقعیت (Fact ) را از بیرون از سیستم دریافت می کند و با توجه به آن واقعیت با پاسخ و راه حل مناسب (خبرگی) را به عنوان خروجی می دهد. این سیستم در حالت کلی از 2 قسمت تشکیل شده است:

پایگاه دانش Knowledge Base

موتور استنتاج Inference Engine

fact

پایگاه دانش Knowledge Base

موتور استنتاج Inference Engine

**خبرگی**

**واقعیتfact**

واقعیات همان اطلاعاتی است که به عنوان ورودی به سیستم خبره داده می شود.

**قواعدrule**

ساختار دانشی است که یعنی اطلاعات دانسته به اطلاعات دیگری می تواند به معلومات قبلی اضافه و یا دانسته فرض شوند.ساختار قاعده یا چند پیش فرض در قسمت "اگر" را با نتیجه گیری ها در قسمت "آنگاه" مرتبط می سازد. یک قاعده می تواند "در غیر این صورت" هم داشته باشد.

در واقع قاعده ساختاری است که با برقرار شدن قسمت الگو "pattern" قسمت عمل “action” اجرا می شود که اگر این عمل خودش معادل یک الگو دیگر شود عمل مربوط به آن اجرا می شود و به همین ترتیب تا آخر ادامه پیدا می کند.

If pattern

Then action

Else action

پایگاه دانش

Step 3

Rule 1) if Ball’s color is red

Then I like it

Rule2) if I like it

Then I will buy it

Step 5

Step 2

Ball’s color is red (Fact 1)

I like the ball (Fact 2)

I will try

Step 4

Ball’s color?

Red

Step 1

حافظه کاری Working memory

در اين مثال واقعيات(Facts) در حافظه كارى و قواعد ‎(Rules) ‏ در پایگاه دانش قرار دارند.قاعده ‎Rule1‏ اجرا مى شود و قسمت الگوی ‏آن Fact 1‎ ‏ تطبيق دارد یعنی ‎(Ball's Color Is Red) ‏ پس چون قسمت الگو با ‎Fact1‏ تطبيق داشت قسمت عمل ‎Rule1 ‏ يعنى I like It‏ انجام مى شود. سپس Rule2اجرا مى شود جون الگوی آن با Fact 2كه اين قسمت توسط عمل l‏‎ Rule یعنی I like it ساخته شده است تطبيق دارد قسمت عمل ‎Rule2‏ يعنى ‎Twill Buy It‏ اجرا می شود.

**نرم افزار کلیپس ‎(CLIPS)**‏

‏از اين نرم افزار براى ساخت سيستم هاى خبره استفاده مي شود.

برخى از ویژگی های نرم افزار به شرح زير مى باشد.

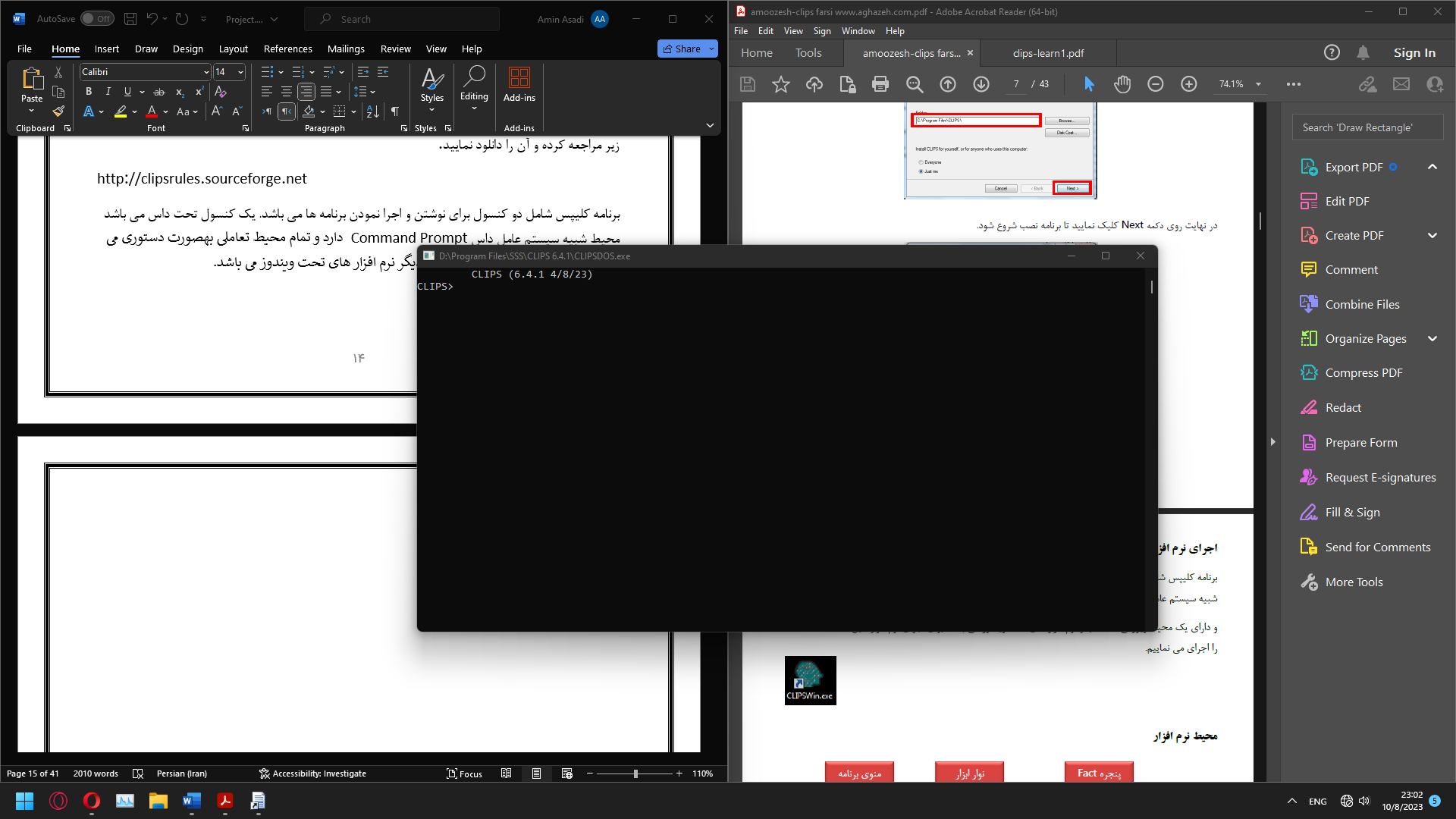
* از سر نام هاى عبارات ‎Language Implementation Production System‏ C كرفته شده است
* براى اولين با توسط ‎NASA‏ در واحد فناورى نرم افزار(‎(Software Technology Branch ‏ توليد شده است .
* اين نرم افزار رایگان بوده و در حال حاضر نسخه ‎6.3‏ آن موجود مى باشد.
* اين نرم افزار بر روى سيستم عامل هاى متفاوت مانند ويندوز لينوكس. مك نصب مى گردد.
* اين نرم افزار با استفاده از زبان C ساخته شده است.
* اين نرم افزار چند كاره مى باشد يعنى مى توانيد بصورت مدل هاى زير با آن برنامه نويسى كنيد.
* بر مبناى قواعد ‎(Rule-based) ‏
* ‏ بر مبناى شی گرایی ‎(Object-oriented)‏
* بر مبناى روال ها‏ ‎(Procedural programming) ‏
* نرم افزار كليبس فقط قواعد زنجيره اى بيشرو را يشتيبانى مى كند.

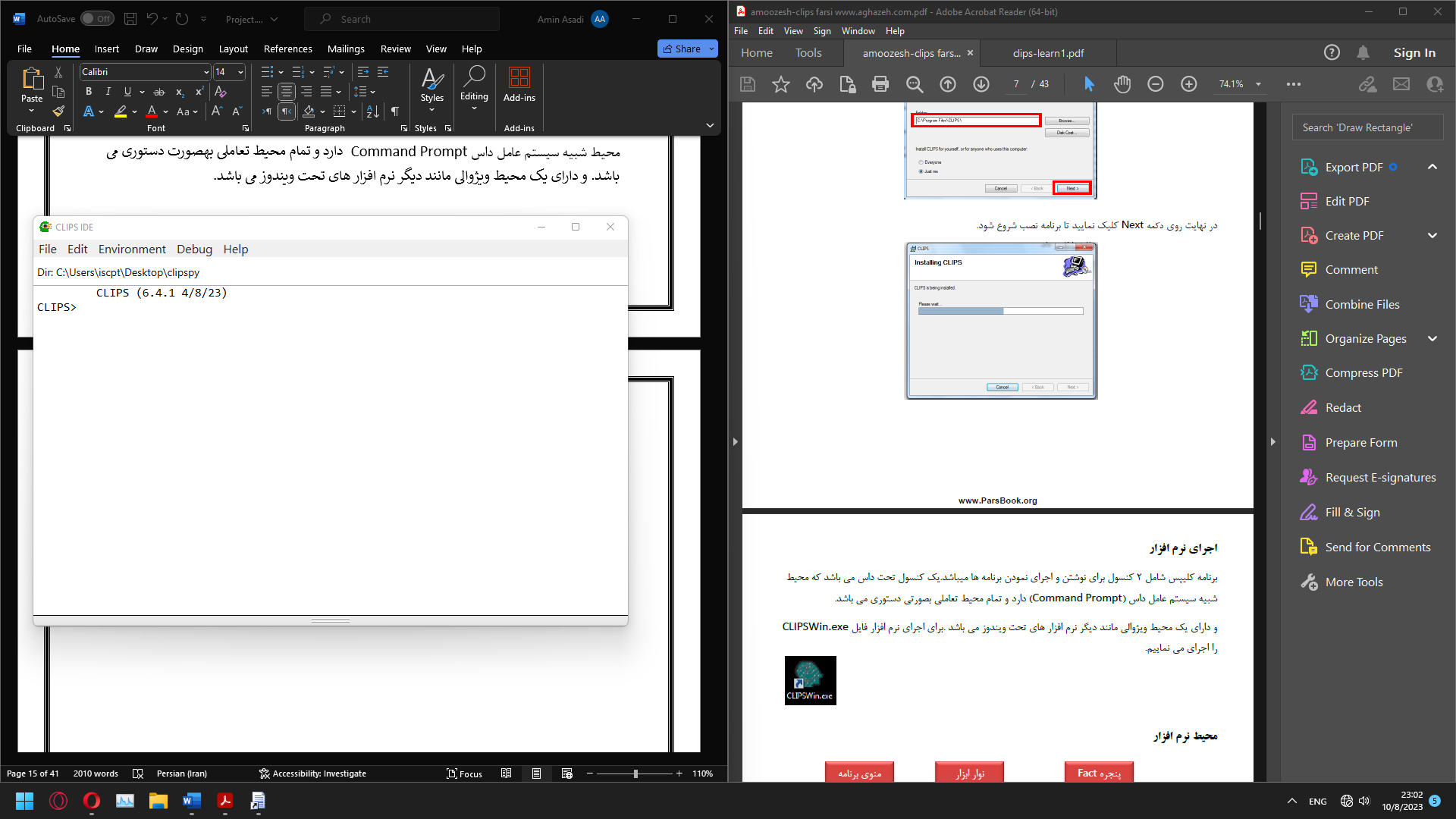
دستيابى به نرم افزار

اين نرم افزار یک برنامه رایگان مى باشد كه هم اكنون نسخه 6.3 آن موجود مى باشد مى توانيد اين نرم افزار ‎را‏ براى سيستم عامل هاى موجود يعنى ويندوز. لينوكس و مك تهيه نماييد به همين منظور مى توانيد به سايت زير مراجعه كرده و آن را دانلود نمایید.

‎http://clipsrules.sourceforge.net

برنامه کلیپس شامل دو کنسول برای نوشتن و اجرا نمودن برنامه ها می باشد. یک کنسول تحت داس می باشد محیط شبیه سیستم عامل داس Command Prompt دارد و تمام محیط تعاملی بهصورت دستوری می باشد. و دارای یک محیط ویژوالی مانند دیگر نرم افزار های تحت ویندوز می باشد.





مولفه های پایه نرم افزار

* فیلدfield
* واقعیاتfact
* قواعدrule
* الگوtemplate
* کلاسclass

**فیلدها**

برای‌ ساخت‌ پایگاه دانش‌ باید اطلاعات از ورودی‌(صفحه‌ کلید یا فایل‌) خوانده شود سپس‌ دستورات اجرا شـوند.در طی‌ فرایند اجرای‌ برنامه‌، نرم افزار کلیپس‌ نشانه‌ ها Symbol،کاراکتر هایی‌ که‌ یک‌ معنی‌ می‌ دهد را با یگدیکر ترکیب‌ و گروه بندی‌ می‌ کند و یک‌ توکن‌ می‌ سازد.در واقع‌ فیلد یک‌ نوع خاصی‌ از توکن‌ می‌ باشد.

**فیلد عددي**

دو نوع فیلد عددی‌ وجود دارد.

عدد صحیح‌ Integer مانند ٢٣٨ ، -٣٢

عدد ممیز شناور Float مانند ٣E٢٧٣، ٠٢.١٥ ، ٧.E٢٥E- همان نماد علمی‌ می‌ باشد.

**فیلد نشانه‌ اي**

• فیلد های‌ نشانه‌ ای‌ حاوی‌ کاراکتر های‌ اسکی‌ می‌ باشند.

• این‌ فیلد ها دارای‌ محدودیت‌ هایی‌ می‌ باشند.

• نشانه‌ ها در کلیپس‌ حساس به‌ حروف Case Sensitive هستند.

• نشانه‌ ها نمی‌ توانند با کاراکتر های‌ ؟ و ؟$ شروع شوند چون این‌ ٢ کاراکتر برای‌ معرفی‌ متغیر ها رزرو شده اند.

• نمونه‌ هایی‌ از نشانه‌ ها foo: ،

**محدودیت‌ فیلد نشانه‌ اي**

فیلد های‌ نشانه‌ ای‌ نمی‌ توانند شامل‌ سمبل‌ های‌ زیر باشند.

• کاراکتر های‌ اسکی‌ چاپ نشدنی‌(مانند کاراکتر فاصله‌(((Space

• دابل‌ کوتیشین‌ ” ”

• پرانتز با و بسته‌ ( )

• امپرسند &

• خط‌ عمودی‌ |

• علامت‌ کوچکتر <

• علامت‌ تیلدا ~

• سمی‌ کولن‌ ;

**فیلد رشته‌ اي**

فیلد های‌ رشته‌ ای‌ در بین‌ ٢ دابل‌ کوتیشن‌ ” ” قرار می‌ گیرند.تنها محدودیت‌ در این‌ فیلد ها استفاده از بک‌ اسـلش‌ \ مـی‌ باشـد.که‌ برای‌ چاپ بک‌ اسلش‌ از \\ و برای‌ چاپ ” از \ استفاده می‌ شود.در نمونه‌ هایی‌ آورده شده است‌.

* “foo”
* “a and b”
* “a\”quote”
* “1 number”

فیلد آدرس

این‌ فیلد ها آدرس داده ها را نگه‌ داری‌ می‌ کنند مانند زبان .C

واقعیات((Fact

Fact ها در واقع‌ واقعیت‌ هایی‌ هستند که‌ کاربر به‌ عنوان ورودی‌ به‌ سیستم‌ خبره به‌ می‌ دهد و از سیستم‌ خبـره ،خبرگـی‌ دریافت‌ می‌ کنند. کلیپس‌ برای‌ حل‌ مسئله‌ ها نیاز به‌ اطلاعات و داده هـا دارد.پایـه‌ واحـد هـای‌ داده کـه‌ در قواعـد((Rule استفاده می‌ شود همان Fact می‌ باشد.هر Fact از قسمت‌ های‌ زیر تشکیل‌ شده است‌.

* نام واقعیت‌((Relation name
* تعداد صفر یا بیشتر اسلات((Slot name به‌ همراه مقدار((Slot value



مثال

Person (

(Family “Asghari”)

(Age 24)

)

ساختار :Deftemplate

با استفاده از این‌ دستور می‌ توان ساختار **fact** ها را بصورت گروهی‌ ایجاد نمود.باید دقت‌ نمود که‌ این‌ دستور فقط‌ ساختاراولیه‌ **Fact** را می‌ سازد و هیچ‌ **Fact** را ایجاد نمی‌ کند و بایـد بـا دسـتور **Assert** آن سـاختار ایجـاد شـده توسـط‌ **Deftemplate** را مقدار دهی‌ نماییم‌.

**Syntax:**

**( deftemplate <relation-name> [<optional-comment>]**

**<slot-definition>\* )**

**<slot-definition>**

**(slot <slot-name>) | (multislot <slot-name>)**

(

:**<relation-name>** نام الگو می‌ باشد.

:**<optional-comment>** می‌ توان در این‌ قسمت‌ توضیحاتی‌ برای‌ الگو قرار داد.این‌ قسمت‌ اختیاری‌ می‌ باشد.

:**(slot <slot-name>)** این‌ قسمت‌ اجازه می دهد یک‌ فیلد با یک‌ مقدار ذخیره گردد.

:**(multislot <slot-name>)** این‌ فیلد اجازه می‌ دهد چندین‌ مقدار در یک‌ اسلات ذخیره شود.

در مثال زیر یک‌ فقط‌ ساختار ایجاد شده است‌ و تا زمان مقدار دهی‌ با دستور **Assert** این‌ الگو به‌ **Fact** تبدیل‌ نمی‌ شود.

**(Deftemplate Person “This Template is For one Person”**

**(MultiSlot Name)**

**(Slot Age)**

**(Slot Eye-Color)**

**(Slot Hair-Color)**

**)**

دستور Assert

مقداردهی‌ ساختار Template با دستور Assert انجام می‌ شود.

Example:

(Assert Person(

(Name John B. Jakson)

(Age 25)

(Eye-Color Brown)

(Hair-Color Black)

)

)

با دستور **Assert** می‌ تواند **Fact** هایی‌ بصورت جداگانه‌ نیز تعریف‌ کرد که‌ در این‌ صورت نیـازی‌ بـه‌ تعریـف‌ **Template** نمی‌ باشد.

(Assert(Color Red))

مثال فوق یک‌ **Fact** با نام **Color** با مقدار **Red** به‌ **Fact-List** اضافه‌ می‌ کند.

**(Assert(Color Red)**

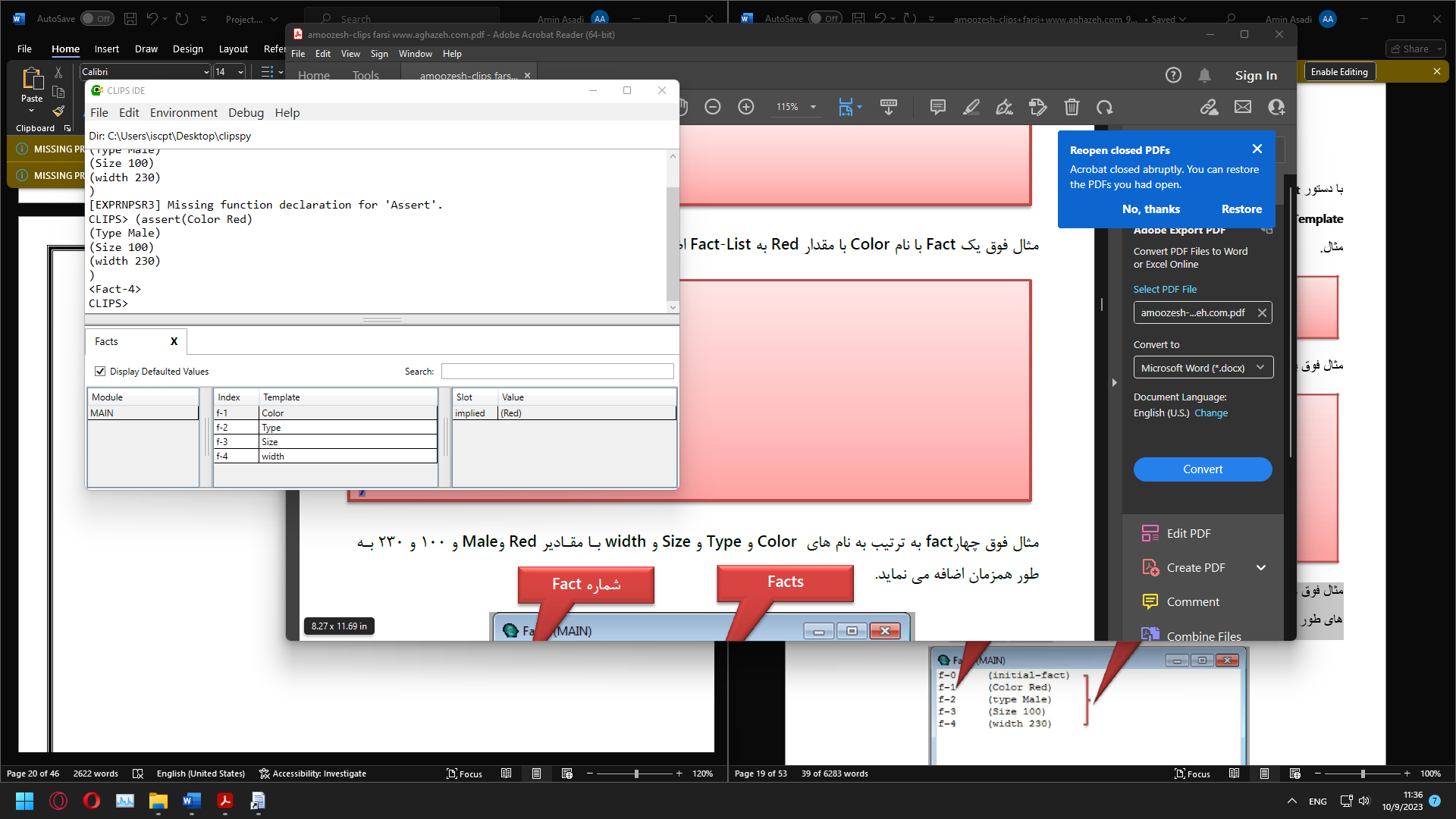
**(Type Male)**

**(Size 100)**

**(width 230)**

**)**

مثال فوق چهارfact به‌ ترتیب‌ به‌ نام های‌ Color و Type و Size و width بـا مقـادیر Red وMale و ١٠٠ و ٢٣٠ بـه‌ طور همزمان اضافه‌ می‌ نماید

.

محدودیت ها برای اسلات

با اعمال محدودیت ها می توان اسلات های یک تمپلیت را مجبور به وارد کردن داده های با قالب خاص نمود و یا مقداری پیش فرض برای اسلات ها در نظر گرفت.

constraint-attribute> ::= <type-attribute > |

< allowed-constant-attribute | >

< range-attribute | >

< cardinality-attribute >

< default-attribute >

) < type-attribute> ::= (type <type-specification ) >

< <type-specification> ::= <allowed-type>+ | ?VARIABLE

< allowed-type> ::= SYMBOL | STRING | LEXEME |

INTEGER | FLOAT | NUMBER |

INSTANCE-NAME | INSTANCE-ADDRESS |

INSTANCE | FACT-ADDRESS |

EXTERNAL-ADDRESS

<allowed-constant-attribute>

::= (allowedsymbols <symbol-list>) |

(allowedstrings <string-list>) |

(allowed-lexemes <lexeme-list>) |

(allowedintegers <integer-list>) |

(allowedfloats <float-list>) |

(allowednumbers <number-list>) |

(allowed-instance-names <instance-list>) |

(allowedvalues <value-list>)

مثال

deftemplate person

نام فقط‌ می‌ تواند نوع

رشته‌ ای‌ قبول نماید

(slot name (type STRING))

(slot age (type INTEGER))

سن‌ فقط‌ می‌ تواند نوع

عدد صحیح‌ قبول نماید

(slot gender (allowed-symbols male female))

)

جنیست‌((Gender فقط‌ می‌ تواند شامل‌ یکی‌ از ٢ مقدار”Male” يا ”Female” باشد.

به‌ عنوان مثال در صورتی‌ که‌ سعی‌ کنید مقداری‌ بر خلاف محدودیت‌ تعریف‌ شده وارد نمایید با خطایی‌ از سوی‌ نرم افـزار کلیپس‌ مواجه‌ خواهید شد.

متغیر ها Variable

|  |  |  |
| --- | --- | --- |
| در نرم افزار کلیپس‌ میتوان ٢ نوع متغیر تعریف‌ نمود. | | |
| • | متغیر تک‌ مقداری‌((**Single Value ?<Symbol>** |
| • | متغیر چند مقداری‌((**Multi Values $?<Symbol>** |

ویلکارد ها((Wildcard

ویلکاردها علامت‌ هایی‌ هستند که‌ بجای‌ هر کاراکتر یا کاراکتر خاصی‌ بکار می‌ روند.مانند \* که‌ در هنگام جستجو به‌ معنای‌ ”هر” بکار می‌ رود.

بدست‌ آوردن شماره ایندکس‌ Fact

برای‌ انجام بعضی‌ از عملیات ها نیاز به‌ شماره ایندکس‌ **Fact** می‌ باشد به‌ همین‌ دلیل‌ از روش زیر شماره ایندکس‌ را بدست‌ می‌ آوریم‌.باید بدانیم‌ که‌ شماره ایندکس‌ منظور همان شماره ای‌ است‌ که‌ در پنجـره **Fact-List** در سـمت‌ چـپ‌ **Fact** نشان داده می‌ شود و این‌ شماره را خود نرم افزار کلیپس‌ بصورت اتوماتیک‌ به‌ **Fact** ها اختصاص می‌ دهد.

<variable-symbol> <- <pattern-CE>

:**<variable-symbol>نام یک‌ متغیر است‌ که‌ شماره ایندکس‌ در آن قرار می‌ گیرد.**

**:<pattern-CE> نام Fact است‌ که‌ می‌ خواهیم‌ شماره ایندکس‌ آنرا بدست‌ بیاوریم‌.**

**: <- علامت‌ انتساب می‌ باشد.**

f-0 (initial-fact)

f-1 (data 1 blue)

f-2 (data 1 blue red)

f-3 (data 1 blue red 6.9)

در لیست‌ بالا تعداد چهار **Fact** وجود دارد شماره ایندکس‌ 0 یعنی‌ **f-0** که‌ توسط‌ خود نرم افزار کلیپس‌ رزرو می‌ شود و در واقع‌ یک‌ مقدار دهی‌ اولیه‌ انجام می‌ دهد حتی‌ اگر هیچ‌ **Fact** تعریف‌ نشده باشد این‌ شماره به‌ عنـوان **f-0 (initial-fact)** توسط‌ خود نرم افزار استفاده می‌ شود و **Fact** ها از شماره ١ به‌ بعد شماره گذاری‌ می‌ شوند.

حذف fact با دستور Retract

با استفاده از این‌ دستور می‌ توان یک‌ **Fact** را حذف نمود.باید دقت‌ نمود که‌ این‌ دستور با داشتن‌ شماره ایندکس‌ **Fact** می‌ تواند آنرا حذف نماید که‌ از روش قبل‌ شماره ایندکس‌ را بدست‌ می‌ آوریم‌.

**(retract <retract-specifier>+ | \*)**

:**<retract-specifier>**شماره ایندکس‌ **Fact** ای‌ را که‌ قرار است‌ حذف نماییم‌، مشخص‌ می‌ کنیم‌.

دستور Save-Fact جهت‌ ذخیره Factها

با استفاده از این‌ دستور می‌ توان کلیه‌ **Fact** های‌ موجود **Fact**) ها در پنجره (**Fact** را در داخل‌ فایل‌ ذخیره نمود.

دستور Load-Facts جهت‌ باز نمودن Fact ها

با استفاده از این‌ دستور می‌ توان **Fact** هایی‌ که‌ قبلا توسط‌ دستور **Save-Facts** در فایل‌ ذخیره نموده ایم‌ را بـه‌ لیست‌ **Fact-list** اضافه‌ نماییم‌.

**تعریف‌ قواعد((Rules**

با این‌ دستور می‌ توان قواعد را تعریف‌ نماییم‌. نرم افزار کلیپس‌ یک‌ برنامه‌ **Rule-based** نیز می‌ باشـد یعنـی‌ بـر مبنای‌ قواعد نیزکار می‌ کند.

(defrule <rule-name> [<optional comment>]

<patterns>\*

=>

<actions>\*

)

:**<rule-name>**این‌ قسمت‌ شامل‌ نام قاعده یا **Rule** می‌ باشد.

:**<optional comment>**در این‌ قسمت‌ می‌ توان توضیحاتی‌ در مورد آن قاعده نوشت‌ .توضیحات اختیاری‌ می‌ باشد.

:**<patterns>**این‌ قسمت‌ الگو را مشخص‌ می‌ نماید یعنی‌ در واقع‌ همان قسمت‌ الگو عبارت شرطی‌ می‌ باشد.

:**<actions>**این‌ قسمت‌ اعمال و دستوراتی‌ می‌ باشد که‌ در صورت برقرار شدن((**True** قسمت‌ **Pattern** اجرا می‌ شود. نکته‌:علامت‌ \* در **Syntax** ها به‌ معنی‌ تعداد ٠ یا بیشتر می‌ باشد**<patterns>\*** یعنی‌ چندین‌ الگو می‌ توان تعریف‌ نمود.

(Defrule Rule1

(car(color red))

=>

(assert(Action ok))

)

یعنی‌ اگر **fact** با نام **car** مقدار فیلد **Color** آن **Red** بود سپس‌ یک‌ **Fact** با دستور **Assert**به‌ نام َ**Action** ایجاد کـن‌ و مقدار آنرا **ok** قرار بده.

نکته‌:علامت‌ **=>** معادل کلمه‌ کلیدی‌ **Then** در شبه‌ کد می‌ باشد

ایجاد Fact گروهی‌ با دستور Deffacts

با استفاده از این‌ دستور می‌ توان **Fact** هایی‌ بصورت گروهی‌ تعریف‌ کرد و به‌ عنوان دانش‌ اولیه‌ بـا پایگـاه دانـش‌ اضافه‌ می‌ شود.

(deffacts <deffacts-name> [<comment>]

<pattern>\*)

:**<deffacts-name>**نام **fact** را مشخص‌ می‌ کند.

نکته‌:**Fact** هایی‌ که‌ با دستور **Deffact** تعریف‌ می‌ شوند در پنجره **Facts** نمایش‌ داده نمی‌ شوند.

:**[<comment>]**توضیحاتی‌ برای‌ **Fact** می‌ باشد که‌ نوشتن‌ این‌ توضیحات اختیاری‌ می‌ باشد.

:**<pattern>**نام الگو یا همان **Fact** ها را مشخص‌ مینماید.

(deffacts BaseFact1

(Color Red)

(type Stuff)

(Height 100)

)

مثال فوق **Fact** با نام **\_aseFact1** شامل‌ فیلد های‌ اولیه‌ **Color,type,Height** را به‌ عنوان دانش‌ اولیـه‌ بـه‌ پایگـاه داده اضافه‌ می‌ نماید.

چاپ رشته‌ / متغیر با دستور Printout

**(printout <logical-name> <print-items>\*)**

:**<logical-name>**در این‌ قسمت‌ می‌ توان نوع خروجی‌ را مشخص‌ نمود بـا کـاراکتر (**Terminal**) **t**مـانیتور بـه‌ عنـوان خروجی‌ در نظر گرفته‌ می‌ شود.

:**<print-items>**در این‌ قسمت‌ متن‌ مورد نظر با آیتم‌ هایی‌ که‌ قرار است‌ چاپ شوند را مشخص‌ می‌ نماییم‌.

نکته‌:عبارت **crlf** تعیین‌ می‌ کند که‌ بعد از چاپ نمودن آیتم‌ ها مکان نما به‌ خط‌ بعدی‌ رود.این‌ کلمـه‌ معـادل **\n** در زبـان **c++** میباشد.

نمایش‌ لیست‌ دستورات

با استفاده از دستورات زیر می‌ توان لیست‌ بعضی‌ از ساختارهایی‌ که‌ تا بحال یاد گرفته‌ ایم‌ را مشاهده نماییم‌. :(**list-defrules**) این‌ دستور لیست‌ قواعد موجود را نمایش‌ می‌ دهد.

:**(list-deftemplates)**این‌ دستور لیست‌ الگو های‌ موجود را نمایش‌ می‌ دهد.

:**(list-deffacts)**این‌ دستور لیست‌ **Fact** هایی‌ راکه‌ با دستور **Deffacts** تعریف‌ کرده ایم‌ را نمایش‌ می‌ دهد.

:(**Facts**)این‌ دستورات لیست‌ تمام **Fact**)**fact** های‌ موجود در پنجره **Facts** که‌ با دسـتورات **assert** یـا **Deftemplate** اضافه‌ نموده ایم‌) را نمایش‌ می‌ دهد.

دستورات اساسی‌

:**(clear)**این‌ دستور تمام **Fact** ها و **Rule** ها را از حافظه‌ کاری‌ حذف می‌ کند معادل این‌ است‌ یکبار از نرم افزار خارج شویم‌ و دوباره وارد شویم‌(**.**(**Shutdown + Restart**

:**(reset)**این‌ دستور اطلاعات مربوط به‌ **Fact** ها را حذف میکند و **Agenda** را ریست‌ می‌ کند.

**:**(**Agenda**) قواعد فعال را نمایش‌ می‌ دهد.

:**(Refresh)** این‌ دستور قواعد را بروز رسانی‌ می‌ نماید.

:(**Run [<Limit>]**)با این‌ دستور قواعد فعال در حافظه‌ کاری‌ اجرا می‌ شود که‌ در قسمت‌ **<Limit>** تعداد قواعـدی‌ را که‌ می‌ خواهیم‌ اجرا شود را مشخص‌ می‌ کنیم‌.مشخص‌ نمودن تعداد اختیاری‌ می‌ باشد.در صورتی‌ که‌ تعداد را مشـخص‌ ننماییم‌ یعنی‌((**Run** در این‌ صورت کلیه‌ دستورات فعال در پنجره **Agenda** اجرا می‌ شوند.

بعضی‌ توابع‌ سودمند

تابع‌ save

با این‌ تابع‌ مجموعه‌ از ساختارهایی‌ که‌ وجود دارند(**Fact** و قواعد و الگوها و ...) را در فایلی‌ ذخیره می‌ نماید.

**(save <file-name>)**

: **<file-name>** مسیر و نام فایل‌ به‌ همراه پسوند آن را مشخص‌ می‌ نماید.

تابع‌ Load

توسط‌ این‌ دستور می‌ توان ساختار هایی‌ که‌ قبلا ذخیره نموده ایم)‌(Save را باز نماید و آماده اجرا کند.

**(load <file-name>)**

: <file-name>مسیر و نام فایل‌ به‌ همراه پسوند آن را مشخص‌ می‌ نماید.

تابع‌ Open

با استفاده از این‌ تابع‌ می‌ توان فایل‌ برنامه‌ کلیپس‌ که‌ قبلا ذخیره نموده ایم‌ را باز نماییم‌.

(open <file-name> <logical-name> [<mode>])

:**<file-name>** این‌ قسمت‌ شامل‌ نام فایل‌ به‌ همراه آدرس کامل‌ و پسوند فایل‌ می‌ باشد.حتما اگر در مسیر علامـت‌ بـک‌ اسش‌ وجود دارد باید آنرا مشخص‌ نماییم‌.

نکته‌:برای‌ مشخص‌ نمودن علامت‌ **\** باید این‌ علامت‌ را ٢ بار پشت‌ سر هم‌ تایپ‌ نماییم‌ تا برای‌ کلیپس‌ معنی‌ **\** ایـن‌ بدهـد یعنی‌ **\\** را تایپ‌ می‌ کنیم‌.

:**<logical-name>** این‌ قسمت‌ یک‌ نام منطقی‌ می‌ باشد که‌ قبلا در برنامه‌ جاری‌ نباید استفاده شده باشد.

:**[<mode>]**این‌ قسمت‌ نحوه دستیابی‌ به‌ فایل‌ را مشخص‌ می‌ نماید.این‌ قسمت‌ اختیاری‌ می‌ باشد.

انواع حالت‌ هاي Mode

"r" read access only

"w" write access only

"r+" read and write access

"a" append access only

در صورتی‌ که‌ تابع‌ **Open** با موفقیت‌ اجرا شود مقدار **True** و در غیر اینصورت مقدار **False** را بر می‌ گرداند.

تابع‌ Close

این‌ تابع‌ بر عکس‌ تابع‌ **Open** کار میکند و فایلی‌ را که‌ قبلا با تابع‌ **Open** باز شده است‌ را می‌ بندد.

**(close [<logical-name>])**

:**[<logical-name>]**نام منطقی‌ فایل‌ است‌ که‌ هنگام **Open** کردن به‌ آن اختصاص دادیم‌.

* اگر تابع‌ **Close** بدون پارامتر فراخوانی‌ شود تمام فایل‌ های‌ باز را می‌ بندد.
* در صورتی‌ که‌ تابع‌ **Open** با موفقیت‌ اجرا شود مقدار **True** و در غیر اینصورت مقدار **False** را بر می‌ گرداند.

تابع‌ Read

با استفاده از این‌ تابع‌ می‌ توان مقداری‌ را از صفحه‌ کلید دریافت‌ نمود یا محتویات یک‌ فایل‌ را خواند.

**(read [<logical-name>])**

:**[<logical-name>]نام منطقی‌ فایل‌ است‌ که‌ هنگام Open کردن به‌ آن اختصاص دادیم‌.این‌ پارامتر اختیاری‌ اسـت‌ و اگر فایل‌ را مشخص‌ نکنیم‌ مقداری‌ را از صفحه‌ کلید دریافت‌ می‌ کند.**

**تابع Bind**

با استفاده از این‌ تابع‌ یک‌ مقدار را به‌ یک‌ متغیر منتقل‌ می‌ نماییم‌.

**(bind <variable> <expression>\*)**

:**<variable>** نام یک‌ متغیر را مشخص‌ می‌ نماید.

:**<expression>**یک‌ عبارت را که‌ قرار است‌ در متغیر قرار گیرد مشخص‌ می‌ نماید.

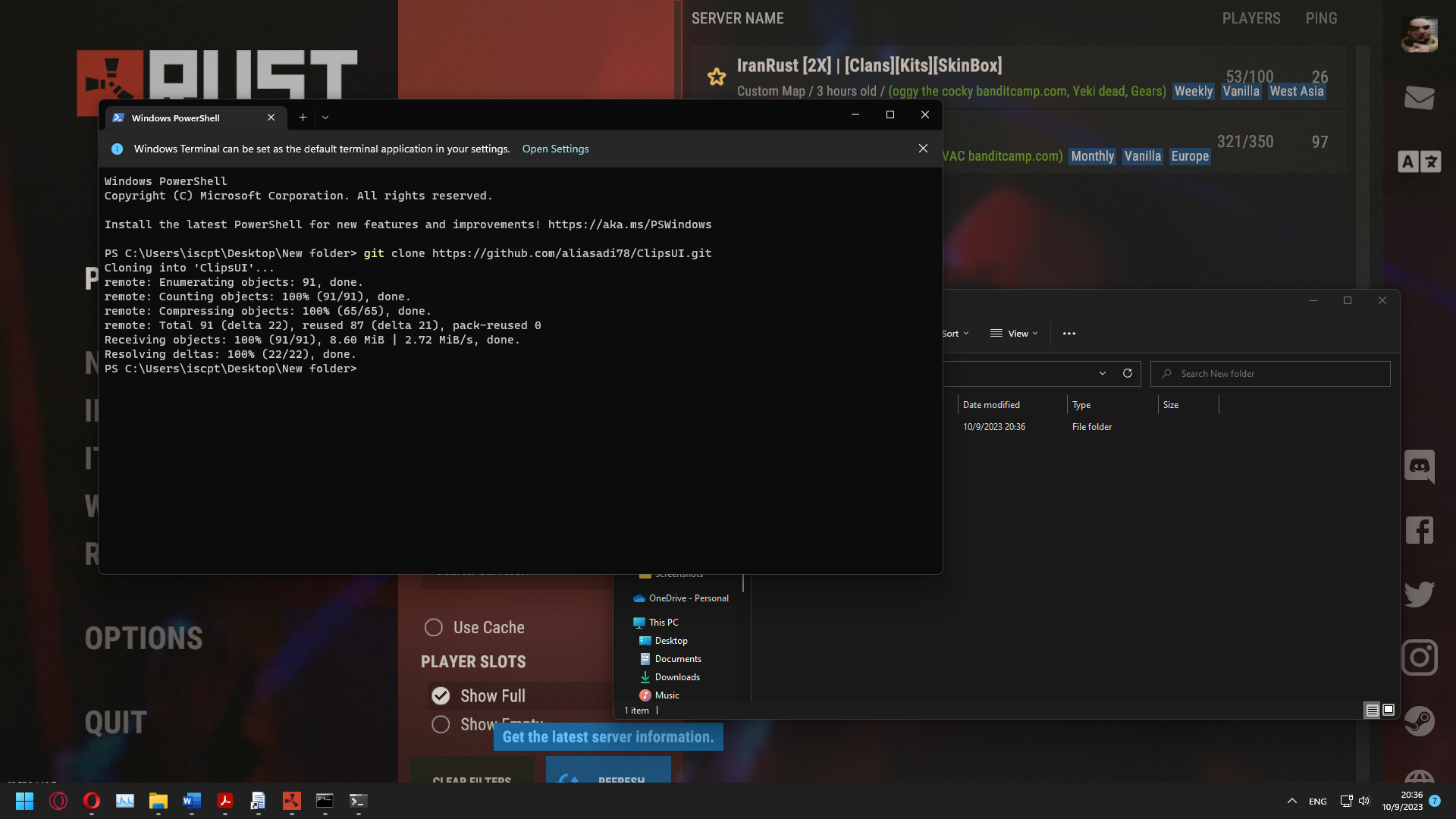
2.راه اندازی برنامه

برنامه کلیپس را از آدرس زیر دانلود کنید

https://www.clipsrules.net

ابتدا با استفاده از دستور زیر پروژه را دانلود می کنیم.

git clone https://github.com/aliasadi78/ClipsUI.git



با دستور cd CLIPSUI وارد پوشه می شویم

برای اجرا پروژه نیاز به نصب Node js داریم اگر روی سیستم نصب دارید با دستور زیر نیازمندی های پروژه را نصب می کنیم.

npm install



و با دستور زیر برنامه رو اجرا می کنیم.

npm start

برنامه در آدرس localhost:3000 اجرا می شود.



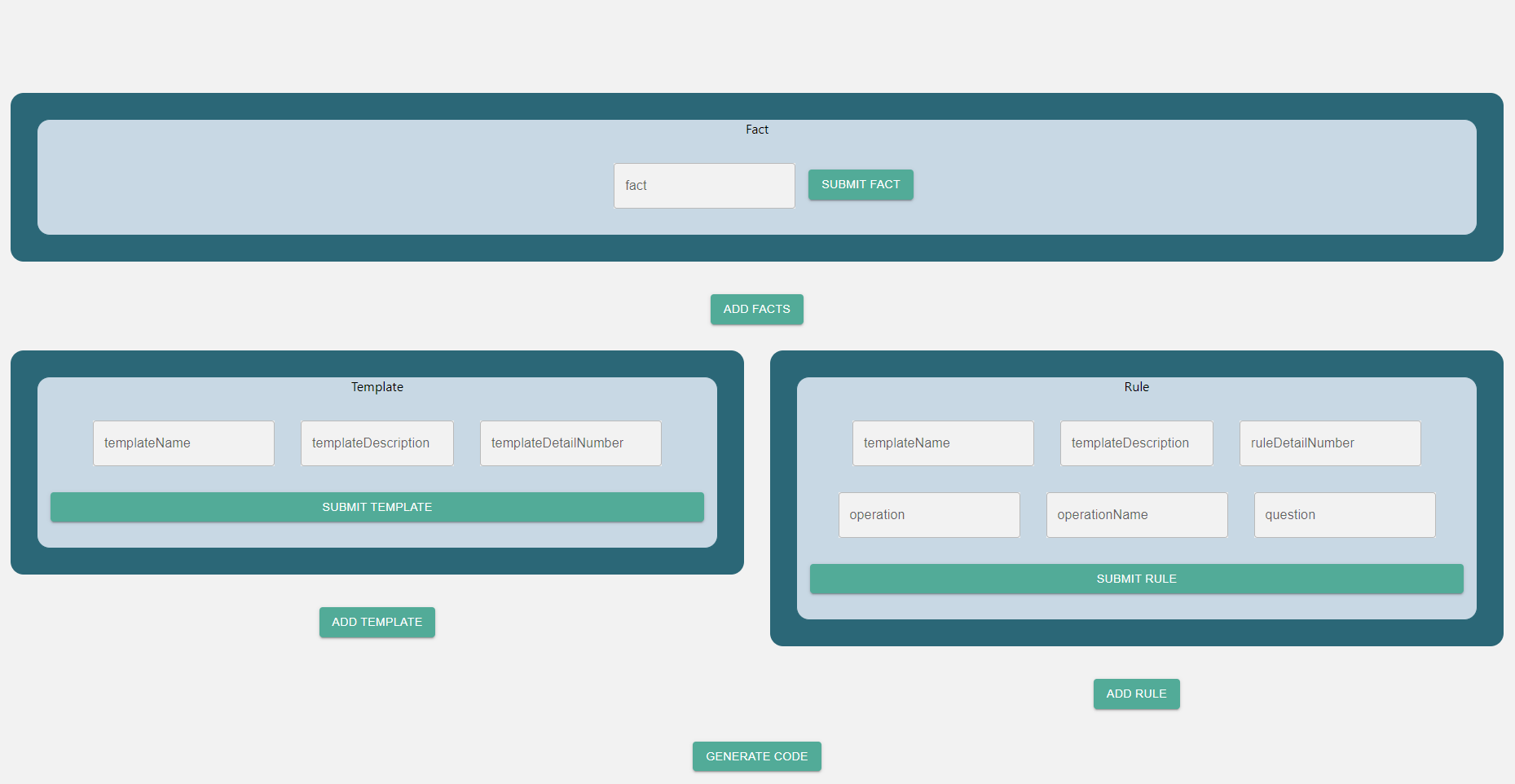
3.نحوه کار برنامه

برنامه از 3 قسمت اصلی تشکیل شده است:

قسمت Fact

قسمت Template

قسمت Rule



برای نوشتن حقایق حقیقت را نوشته و دکمه submit fact را می زنیم تا حقیقت ثبت شود و برای اضافه کردن حقایق بیشتر دکمه add fact را می زنیم.



Template از 3 قسمت نام تمپلیت، توضیح تمپلیت و تعداد جزئیات تشکیل شده است. تعداد جزئیات را وارد می کنیم

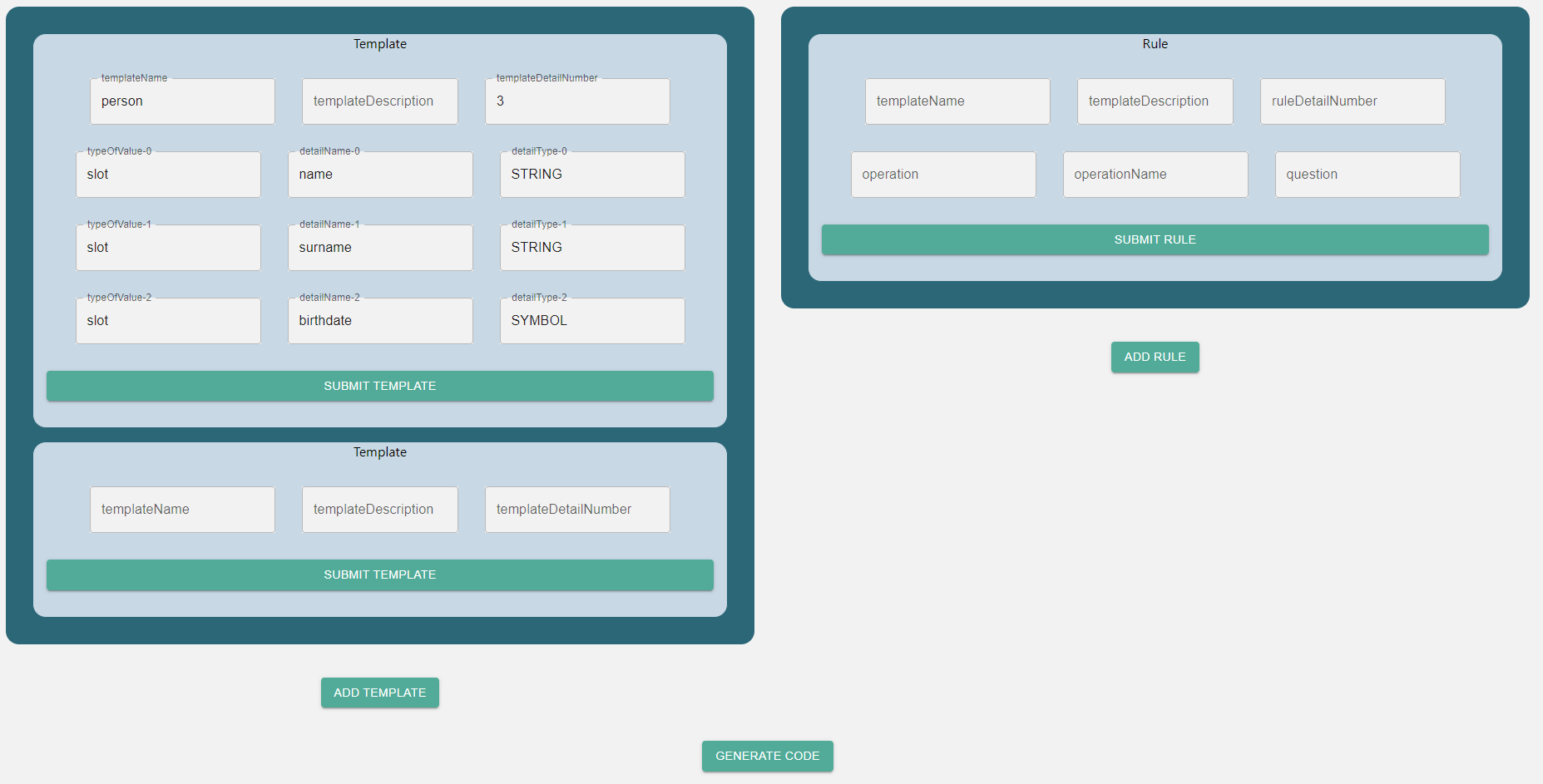
قسمت اول تایپ مقدار را مشخص می کنیم که Slot است یا MultiSlot

قسمت دوم نام را وارد می کنیم

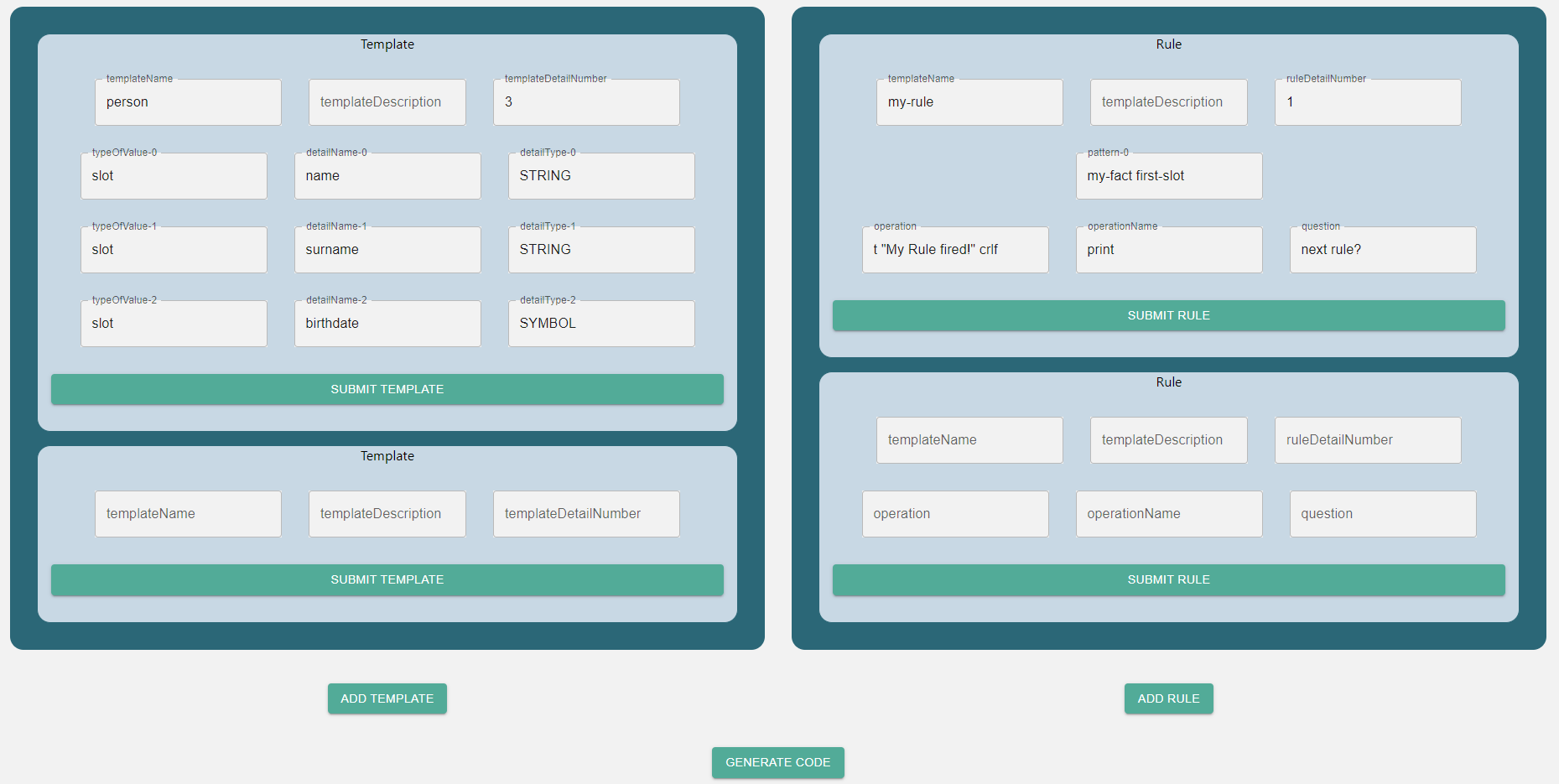
قسمت سوم هم تایپ داده را وارد می کنیم که مقادیر معتبر آن عبارتند از:

|  |
| --- |
| INTEGER |
| FLOAT |
| STRING |
| SYMBOL |
| MULTIFIELD |
| FACT\_ADDRESS |
| INSTANCE\_NAME |
| INSTANCE\_ADDRESS |
| EXTERNAL\_ADDRESS |

مانند قسمت قبل تمپیلت را ثبت و اضافه می کنیم.



Rule از 6 قسمت نام قاعده، توضیح قاعده و تعداد الگوها، عملیات بعد برقراری الگو ها و نام عملیات و قسمت پرسش سوال برای گرفتن ورودی تشکیل شده است. تعداد الگوها و خود الگو ها را وارد می کنیم.



و در آخر دکمه Generate Code را می زنیم تا خروجی را مشاهده کنیم.

4.پیاده سازی یک سناریو

سیستم‌ ساده پزشکی‌

فرض کنید با مطالعه‌ و پژوهش‌ در مورد یک‌ سیستم‌ ساده پزشکی‌ و یا مصاحبه‌ با پزشکان (منظور افراد خبره در این‌ زمینه‌) به‌ اطلاعات زیر دست‌ یافتید:

پزشک‌ برای‌ تشخیص‌ بیماری‌ فرد مراجعه‌ کننده، ابتدا از او چند سوال پرسیده و با توجه‌ به‌ پاسخ‌ های‌ فرد بیمار (که‌ همان حقایق‌ می‌ باشد)، اقدام به‌ تشخیص‌ بیماری‌ می‌ کند.

از جمله‌ این‌ سوال ها این‌ است‌ که‌ **“**محل‌ درد شما کجاست‌؟**“**

مثلا فرد در جواب می‌ تواند بگوید **“**شکم‌**“** ، **“**گلو**“** ، **“**سینه‌**“** و **“**سایر جاها**.“** حال با توجه‌ به‌ جوابی‌ که‌ در قبال این‌ سوال داده می‌ شود (حقیقت‌ تولید شده) مرحله‌ بعدی‌ شروع می‌ شود.

برای‌ هر جواب در این‌ مرحله‌، مراحل‌ بعدی‌ می‌تواند متفاوت باشد. مثلا اگر محل‌ درد **“**شکم‌**“** انتخاب شود، تشخیص‌ داده شود که‌ فرد **“**آپاندیس‌**“** دارد

یا اگر محل‌ درد **“**گلو**“** انتخاب شود در مرحله‌ بعدی‌ از فرد پرسیده شود که‌ آیا **“**تب‌**“** دارد ؟ اگر در جواب این‌ سوال بگوید **“**بله‌**“**، تشخیص‌ داده شود که‌ بیماری‌ **“**گلودرد باکتریایی‌**“** است‌ و اگر **“**خیر**“**، تشخیص‌ داده شود که‌ فرد **“**گلودرد ویروسی‌**“** دارد

اگر محل‌ درد **“**سینه‌**“** انتخاب شود، تشخیص‌ داده شود که‌ فرد **“**سکته‌**“** کرده است‌.

یا اگر محل‌ درد **“**سایر**“** انتخاب شود در مرحله‌ بعدی‌ از فرد پرسیده شود که‌ آیا **“**سرفه‌**“** می‌کند ؟ از در جواب این‌ سوال بگوید **“**خیر**“**، این‌ سیستم‌ ساده نتوانسته‌ بیماری‌ فرد را تشخیص‌ دهد ولی‌ اگر **“**بله‌**“** انتخاب شود در مرحله‌ بعدی‌ از فرد پرسیده شود که‌ آیا **“**تب‌**“** دارد ؟ اگر در جواب این‌ سوال بگوید **“**بله‌**“**، تشخیص‌ داده شود که‌ بیماری‌ **“**آنفولانزا**“** است‌ و اگر **“**خیر**“**، تشخیص‌ داده شود که‌ فرد دچار “سرماخوردگی‌“ شده است‌.

نکته‌: در هر مرحله‌ هر سوال را یک‌ قانون تصور نمود که‌ با توجه‌ به‌ حقایق‌ بدست‌ امده در مرحله‌ قبلی‌ اجرا می‌شود.

درخت تصمیم گیری

شما می توانید از آنچه به عنوان اطلاعات در بخش قبلی به دست آورده ایدیک درخت تصمیم گیری ایجاد کنید. البته توجه کنید ای ن درخت سلسله مراتبی است و از ریشه به سمت برگ ها به جلو می رود.بنابراین در بعضی سیستم ها ترتیب این مراحل می تواند در تصمیم گیری بسیار مهم باشد.

محل درد

هیچکدام شکم

قفسه سینه سینه گلو

سرفه

تب

هیچکدام بله خیر بله

تب

خیر بله

حالا قواعد را مانند شکل وارد برنامه می کنیم.

Menu

=>

t crlf crlf crlf

" mahal dard kojast?lotfan yki az gozinehaye zir ra entekhab konid" crlf crlf

" 1.) shekam. " crlf crlf

" 2.) galo." crlf crlf

" 3.) sineh." crlf crlf

" 4.) sayer." crlf crlf

" 5.) EXIT OF SYSTEM.." crlf crlf crlf

" Your answer: "

(selectedindex)

A-shekam

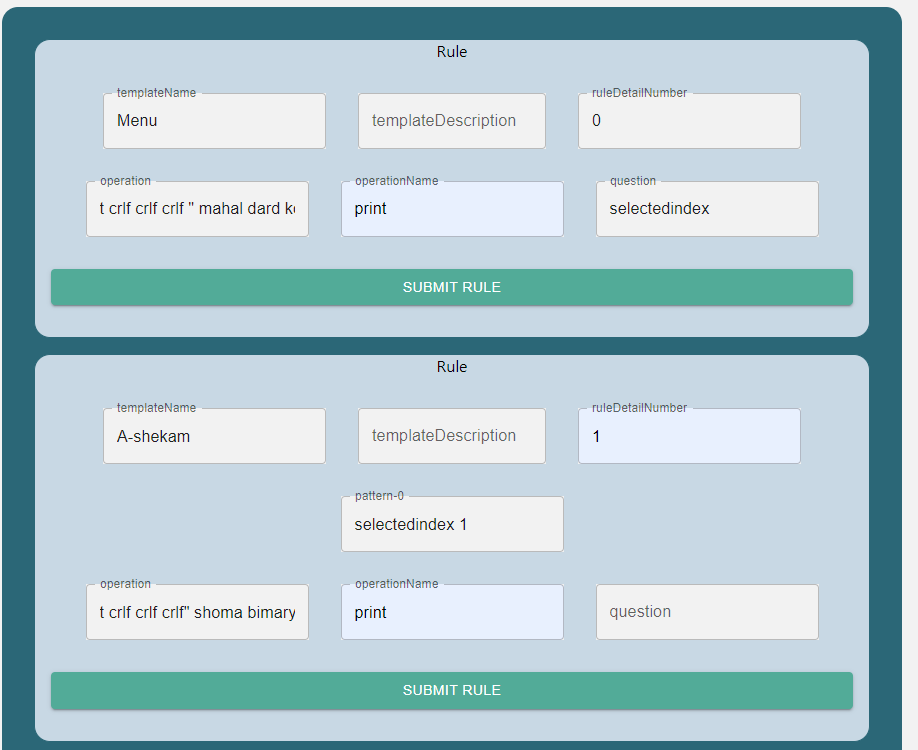
selectedindex 1

=>

t crlf crlf crlf" shoma bimary Apandis darid " crlf

crlf " Thank you for using my Program...

"crlf crlf



Q-galo-tab

selectedindex 2

=>

t crlf crlf crlf " aya shoma tab darid? (Yes | No) " crlf crlf " Your answer: "

ifYesNoTab

A-galo-tab-yes

(selectedindex 2)

(ifYesNoTab yes)

=>

t crlf crlf crlf "shoma glodard bakteraei darid " crlf crlf



A-galo-tab-no

(selectedindex 2)

(ifYesNoTab no)

=>

t crlf crlf crlf "shoma glodard virosi darid " crlf crlf

A-sineh

(selectedindex 3)

=>

t crlf crlf crlf" shoma sekteh kardeid " crlf crlf " Thank you for using my Program... "crlf crlf



Q-sayer-solfeh

(selectedindex 4)

=>

t crlf crlf crlf " aya shoma solfe mikonid? (Yes | No) " crlf crlf " Your answer: "

(ifYesNoSolfeh)

Q-sayer-solfeh-yes-tab ""

(selectedindex 4)

(ifYesNoSolfeh yes)

=>

t crlf crlf crlf " aya shoma tab darid? (Yes | No) " crlf crlf " Your answer: "

(ifYesNoTab)



A-sayer-solfeh-yes-tab-yes ""

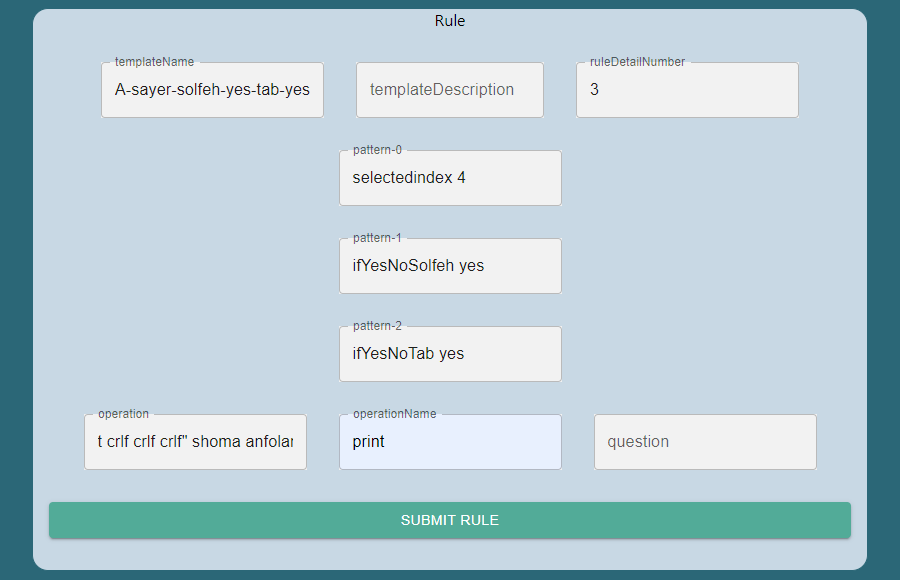
(selectedindex 4)

(ifYesNoSolfeh yes)

(ifYesNoTab yes)

=>

t crlf crlf crlf" shoma anfolanza darid " crlf crlf " Thank you for using my Program... "crlf crlf



A-sayer-solfeh-yes-tab-no ""

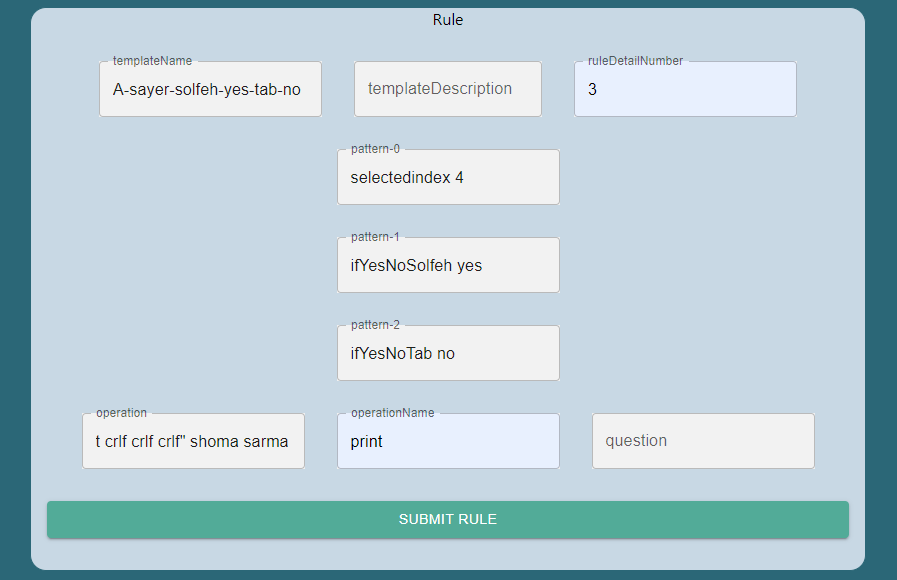
(selectedindex 4)

(ifYesNoSolfeh yes)

(ifYesNoTab no)

=>

t crlf crlf crlf" shoma sarma khordid" crlf crlf " Thank you for using my Program... "crlf crlf



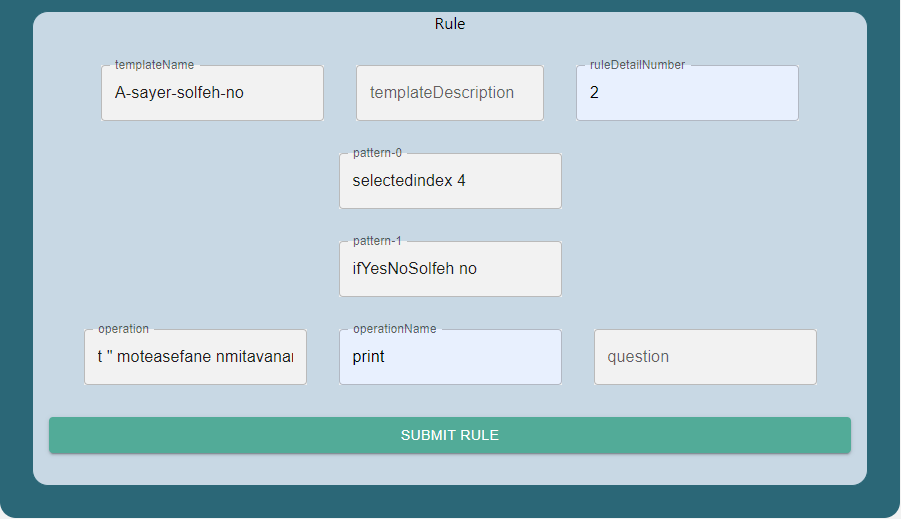
A-sayer-solfeh-no ""

(selectedindex 4)

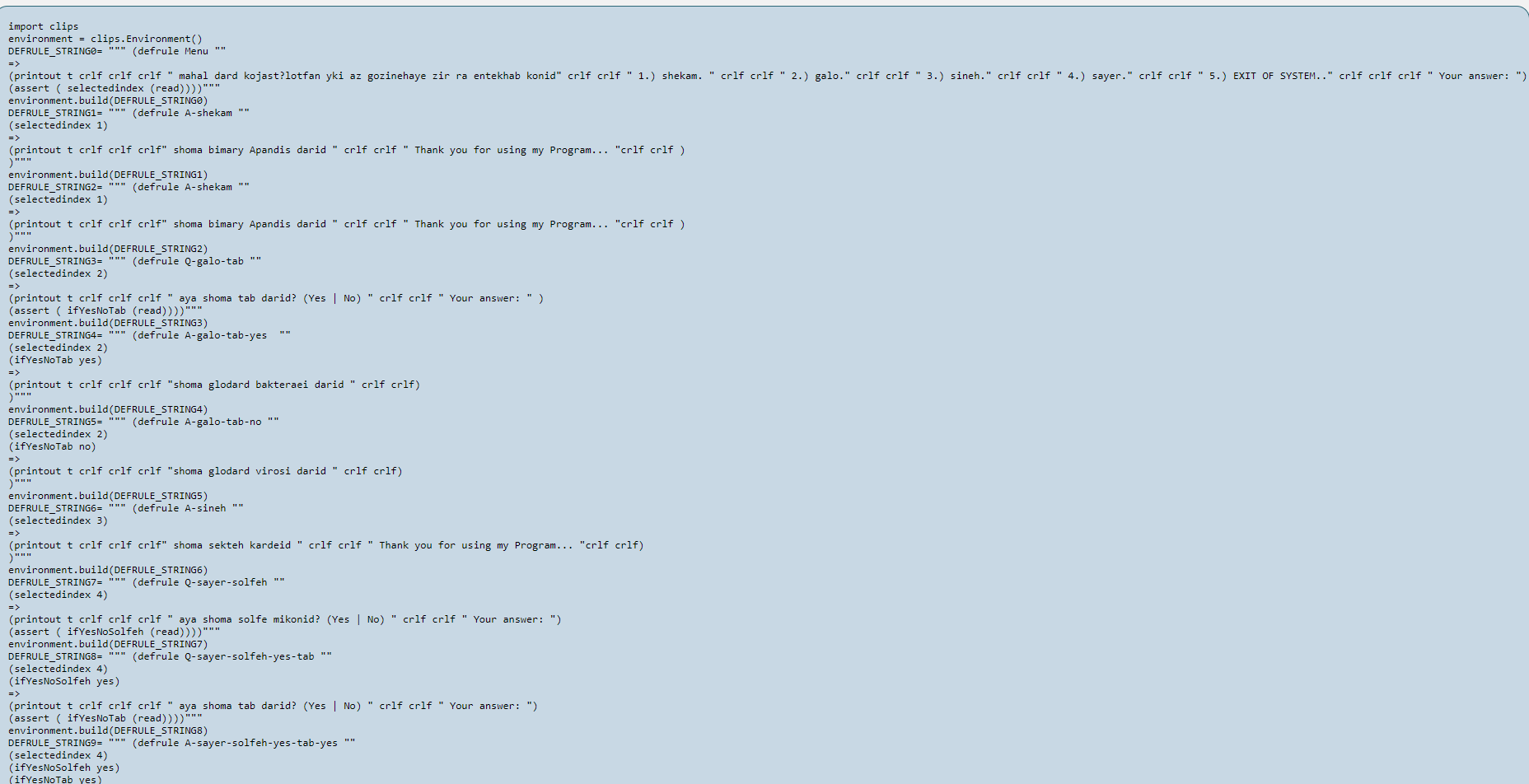
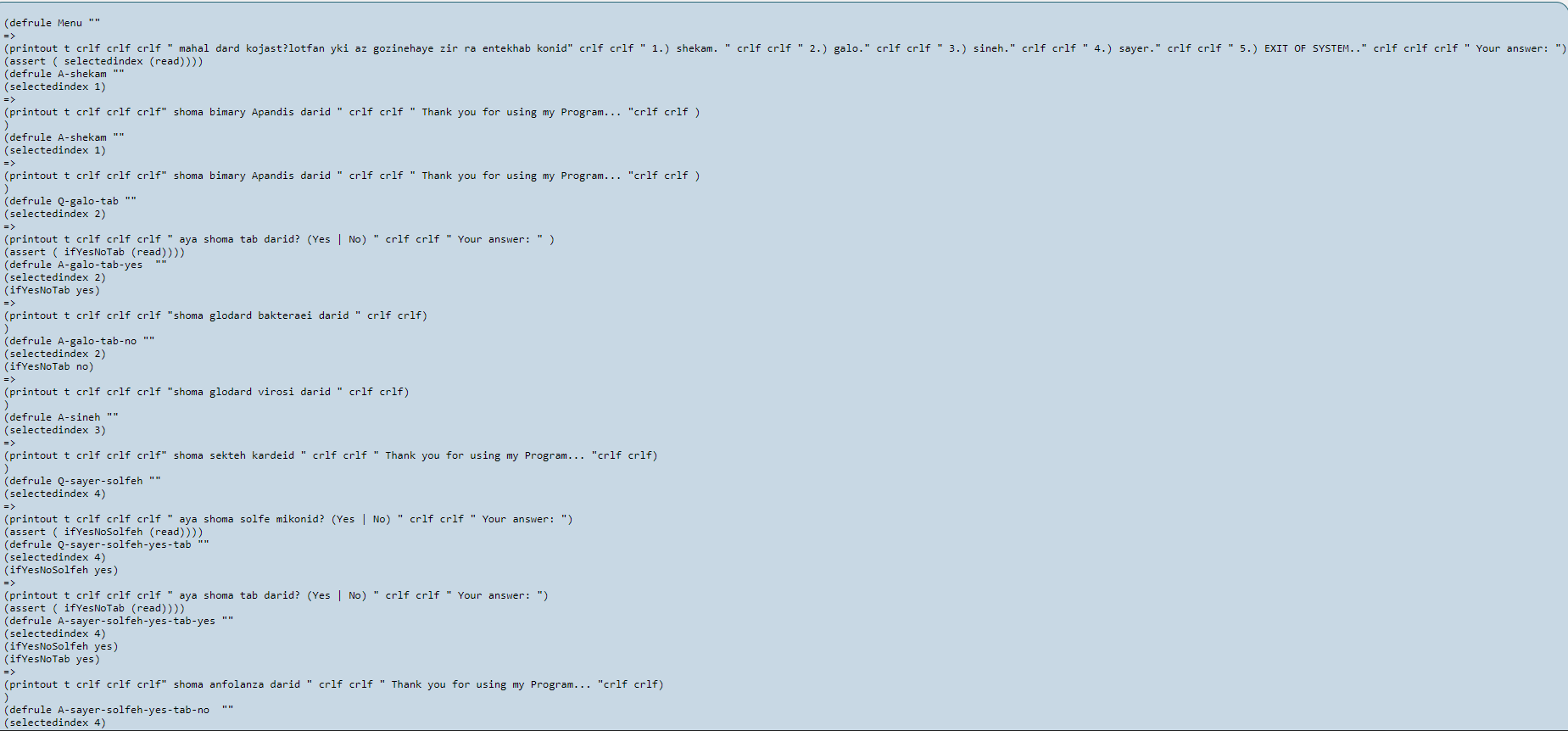
(ifYesNoSolfeh no)

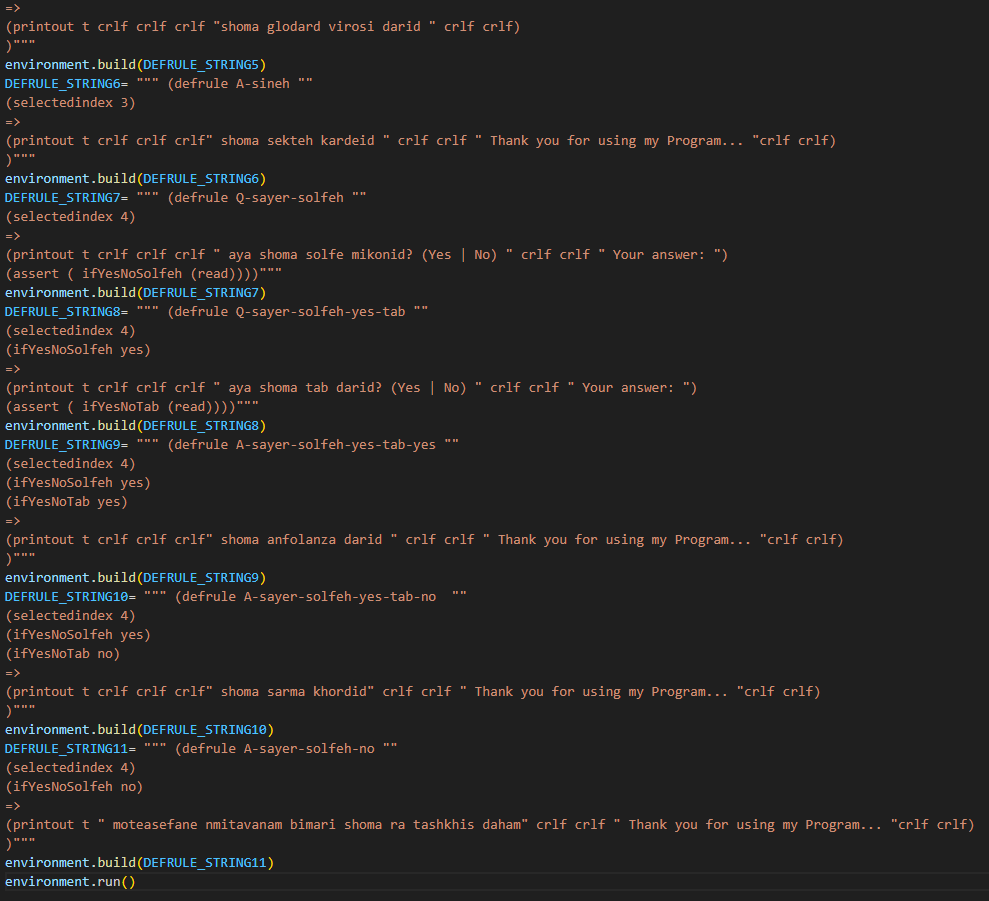
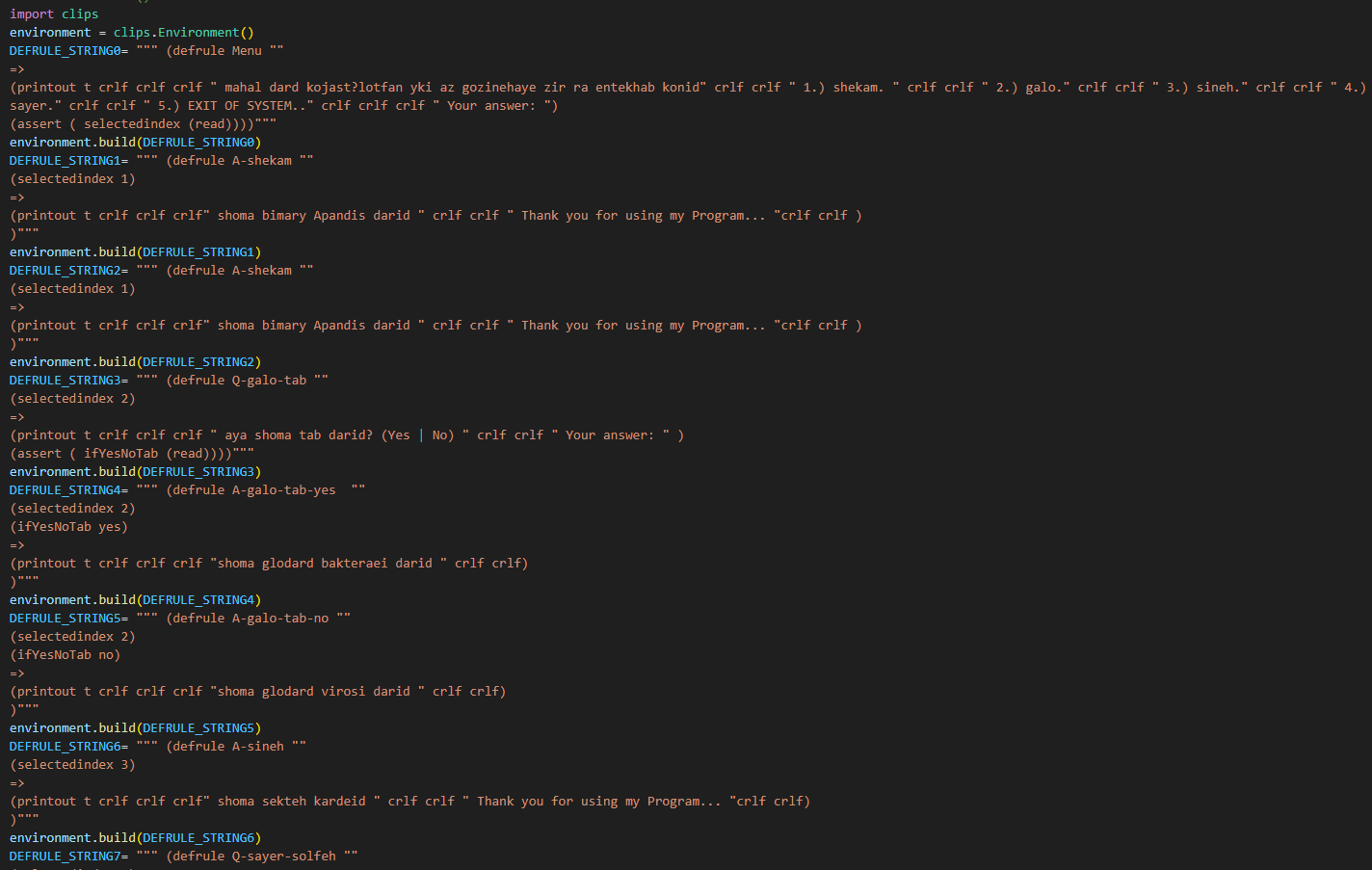
=>

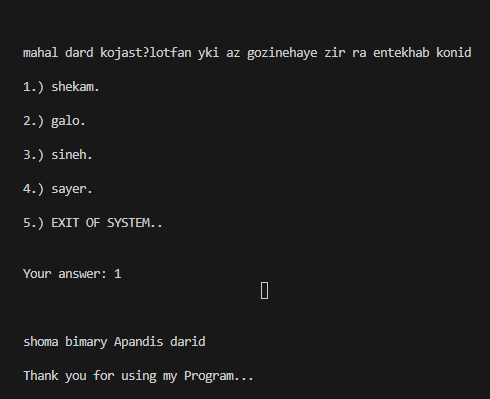
t " moteasefane nmitavanam bimari shoma ra tashkhis daham" crlf crlf " Thank you for using my Program... "crlf crlf



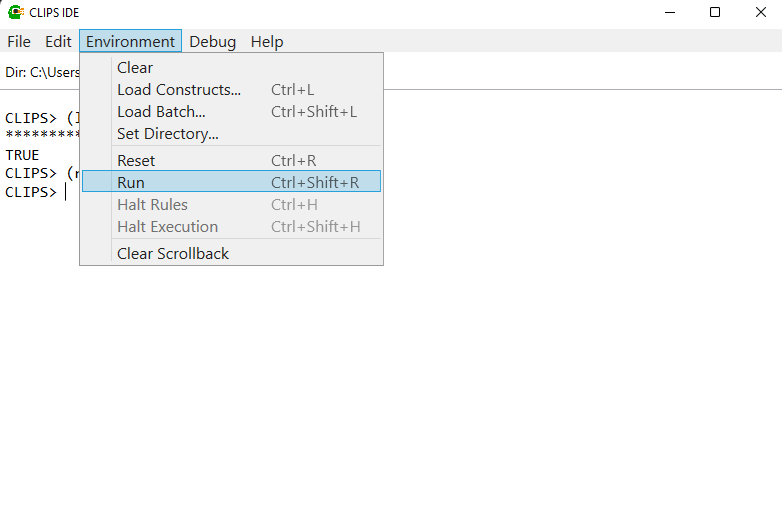
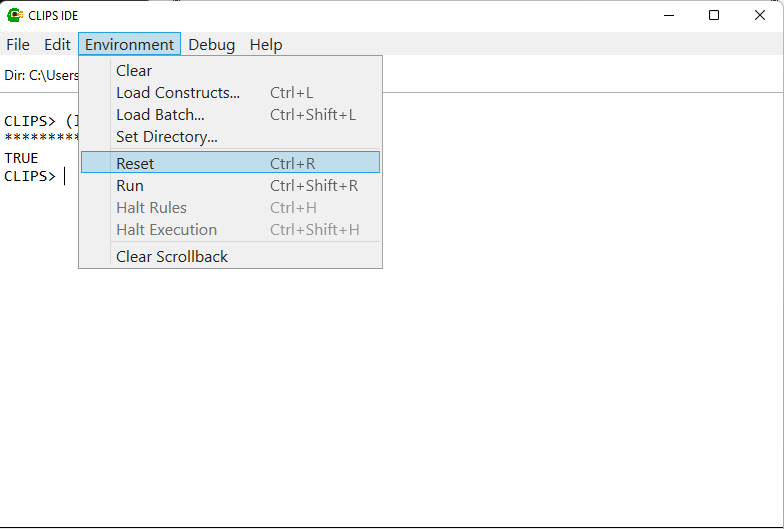
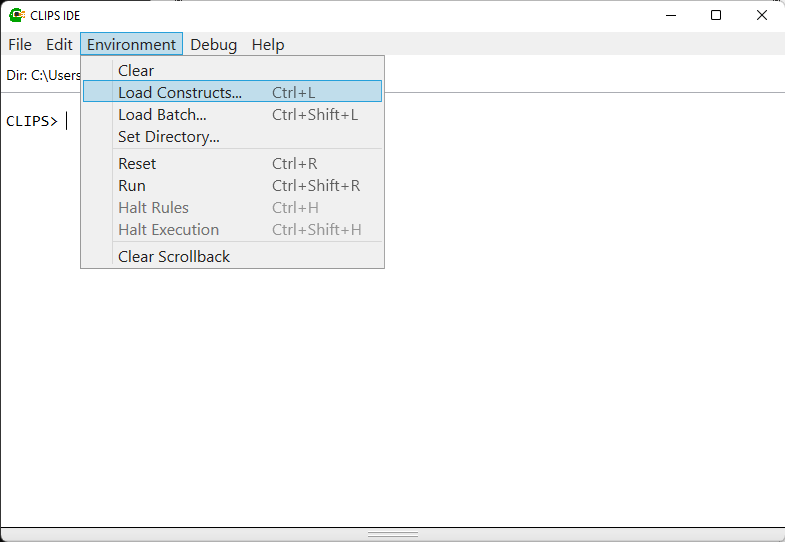
و نتیجه دو کد به زبان های پایتون و کلیپس خروجی می دهد که باید آن ها را اجرا کنیم.

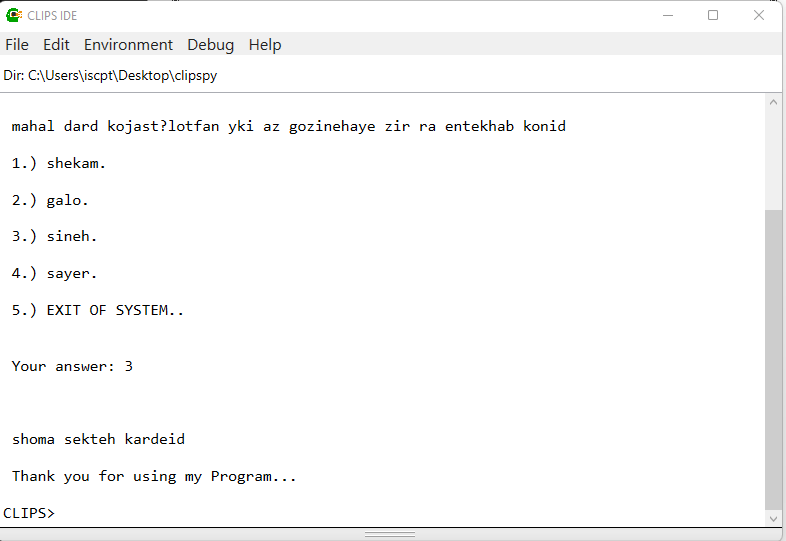
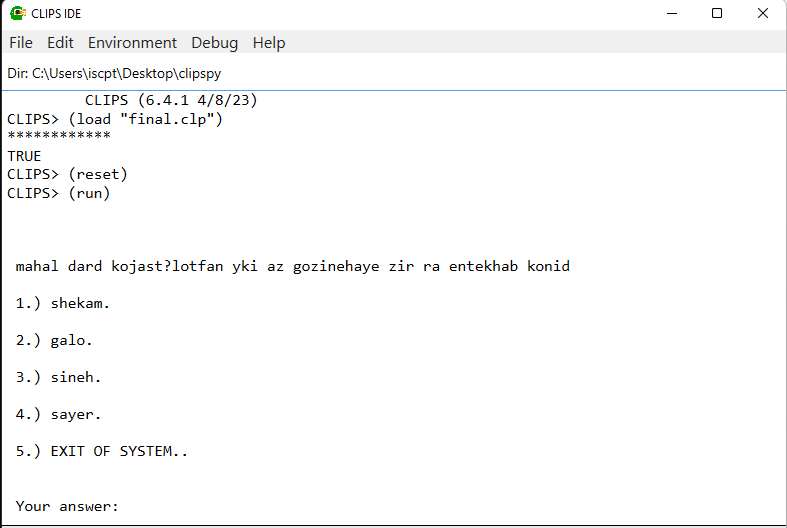






برای اجرای برنامه در کلیپ مانند شکل عمل می کنیم





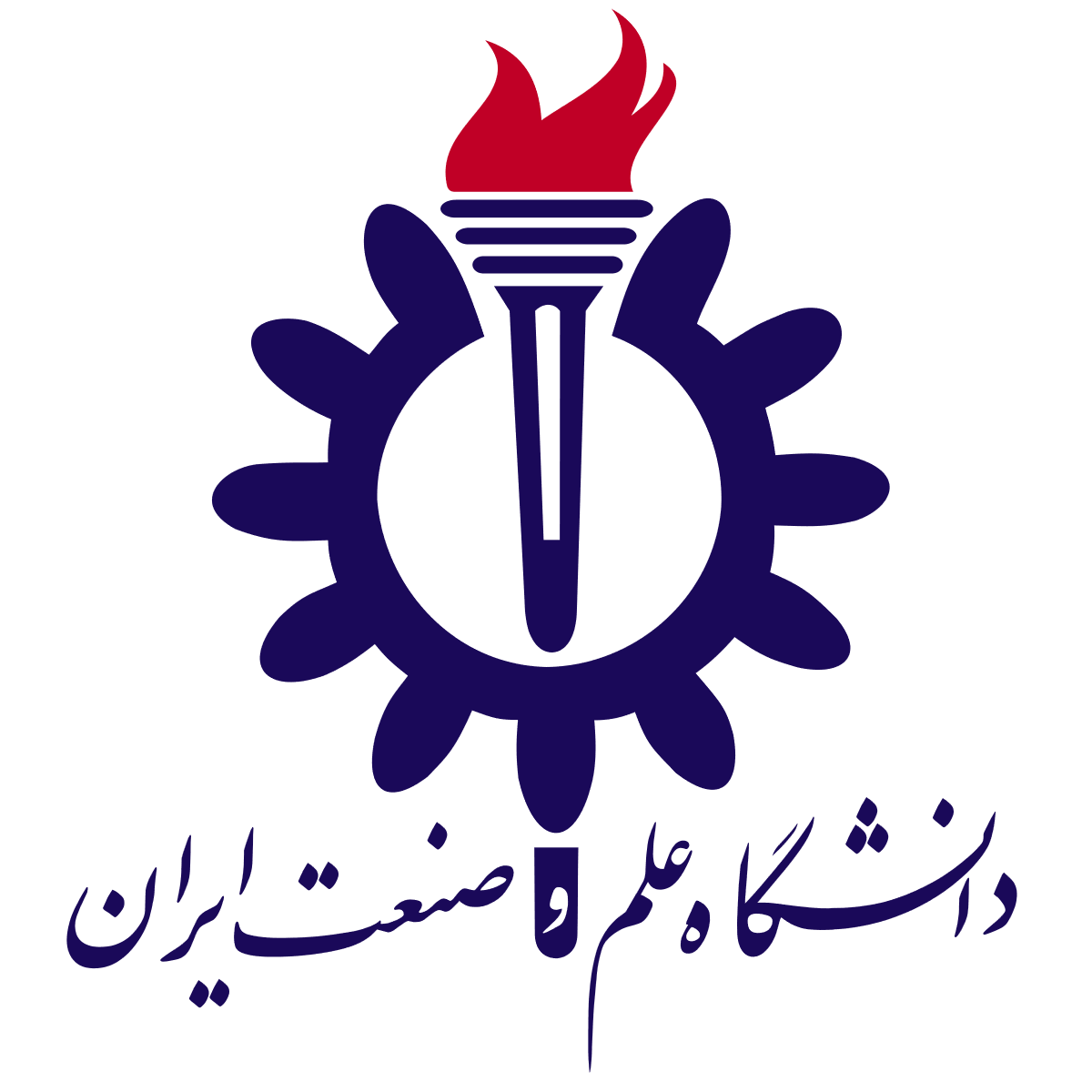
6.منابع و مراجع

<https://www.clipsrules.net>

<https://clipspy.readthedocs.io/en/latest/>

<http://Parsbook.org>

<http://tehranit.net/آموزش-سریع-کلیپس-سیستم-خبره-۴-ساعت/>



Computer Engineering Department

Bachelor's final project

Design and implementation of graphic user interface for Clips software with adaptability feature

Ali Asadi

Supervisor

Dr. Kangavari