



جامعة دمشق

كلية الهندسة المعلوماتية

---

## تقرير تنفيذ مشروع مادة المترجمات

الغروب رقم: A-49.

github: <https://github.com/Nabil762/angular-compiler>

الطلاب: حسان محمد غنام - حازم محمد برهان جاموس - باسل عبد الله الرفاعي

غيث حمزة أبو راشد - نبيل ابراهيم الحمزه ( سنة خامسة ) .

بإشراف : د. باسم قصيبة .

م. آية شحادة.

م. وسيم البذرة.

## هدف المشروع :

-بناء مترجم قادر على التعرف على ملف Angular وبناء شجرة AST والتعرف على الأخطاء

بنوعيتها ( Syntax and semantic errors ).

-تحويل هذا الملف إلى ثلاث ملفات css , html , JS بحيث تعمل على المتصفحات بدون أي مشاكل.

- يمكن أن نضع أكثر من ملف و ربطهم و التفاعل مع الواجهات بشكل بسيط.

## خطوات التنفيذ :

١-بناء ال lexer تم إضافة كل ال key words الموجودة باللغة

المصدر (Angular) بالإضافة للقواعد التي تسمح لنا بتجاوز الفراغات و الأسطر الفارغة والقواعد الأساسية وتم تثبيت التاغات المستخدمة و التي تمت معالجتها بمرحلة الكود جينيريشن لسهولة التعامل معها بمرحلة ال code generation .

لدينا 3 modes :

Mode default , mode template , mode css

٢ - بناء ال parser ترتيب الكلمات (الموجودة بالليكسر) والقواعد بما يتماشى مع قواعد اللغة المصدر (Angular).

تم وضع ال label لجميع القواعد الموجودة أو على الأقل للقواعد التي تحتوي على فقرات عن طريق ال | ووضع اسم لل label بالشكل التالي #LabelName بحيث يجب أن تبدأ بحرف كبير هذا باختصار بما يخص ال lexer و ال parser و ال lable .

٣-بناء ال ast classes بحيث تكون بنيتها بنية مجردة (abstract).

تم بناء كلاس اسمه AstNode نوعه abstract يوجد فيه abstract function generatedCode وحيد لإجبار جميع الكلاسات بعمل @override لهذا ال function ثم بدأنا بإنشاء الكلاسات للقواعد بحيث كل قاعدة موجودة بال parser يجب عمل class لها بالشكل التالي:

إما أن يكون الكلاس كلاس عادي وارث من ال AstNode

أو أن يكون كلاس Abstract وراث من ال AstNode

أو أن يكون الكلاس كلاس عادي وراث من كلاس عبارة عن أب له بال Parser

متى سيكون كلاس عادي و متى يكون abstract

عندما تكون القاعدة تحتوي على | و جميع ال | تحولك ل قاعدة ثانية أي هذا الكلاس لا تحتاج لتخزين أي شيء فيه لذلك سيكون abstract وراث من الكلاس AstNode

أو عندما تكون عبارة عن قاعدة من أحد القواعد ال | بذاتها سيكون كلاس عادي وراث من الأب بالقاعدة .

و إما أن يكون قاعدة ليست من أحد القواعد ال | و لا هو أب يندرج تحته تفرعات للقواعد

فيكون كلاس عادي وراث من الكلاس AstNode

و بالتالي يكون الكلاسات جاهزة يتبقى إضافة ال attribute لكل كلاس مع ال getter & setter

& toString & @override generatedCode

بالشكل التالي : الدخول إسم القاعدة بال parser و معرفة ماذا تريد أن تخزن من معلومات مفيدة مثلاً اسم الكلاس نضع متحول نوعه نصي لتخزين اسم الكلاس أو الباراميتير بالتابع نخزن ليست من الكلاس parameter و هكذا .

أمثلة عن القواعد و تمييز متى يكون الكلاس abstract و متى يكون الكلاس عادي

```
attributes:
    TAG_NAME EQUALH STRING1      # HtmlAttribute
    | DIRECTIVE_NAME EQUALH STRING1 # DirectiveAttribute
    | BINDING_PROPERTY EQUALH STRING1 # BindingAttribute
    | STANDARD_EVENT EQUALH STRING1 # EventAttribute
    | TAG_NAME # HtmlAttribute2
;
```

هنا القاعدة صحيح أنها تحتوي على label ولكن لا توجهنا إلى قاعدة أخرى (كلاس آخر) وإنما نريد إنشاء أوبجكت منه و تخزين ال attribute ضمن الأوبجكت لذلك سيكون كلاس عادي و يرث من الكلاس AstNode.

```

tag:
  standardTag # StandardTagElement
| selfClosingTag # SelfClosingTagElement
;

standardTag :TAG_OPEN (H1| H2| H3| H4| H5| H6| STRONG|UL|P|DIV|LI|BUTTON|FORM|LABEL|INPUT) attribut
selfClosingTag : TAG_OPEN IMG attributes* TAG_SELF_CLOSE;

```

هنا القاعدة tag جميع تفرعاتها عبارة عن توجيه إلى قاعدة أخرى (كلاس آخر) لذلك لا أحتاج إلى تخزين أي شيء و لا إنشاء أو بيجت منه لذلك سيكون Abstract وأبناءه سيرثون منه أي القاعدتين selfClosingTag , standarTag سيرثون من Tag و نوعهم كلاس عادي لحاجتي لإنشاء أو بيجت منه وتخزين المعلومات اللازمة.

```

program: statement+ EOF ;

// ANGULAR
statement: importStatement+ (interfaceDeclaration)* componentDeclaration classDeclaration ;

```

القاعدة program هي أول قاعدة بالبارسر لذلك ترث من الكلاس AstNode و هي كلاس عادي أما القاعدة statment فسترث أيضاً من الكلاس AstNode و كلاس عادي مع أنها تدرج تحت الprogram و لكنه ليس abstract لكي ترث منه.

```

valueExpression:
  STRING # StringValue
| NUMBER # NumericValue
| BOOLEAN # BooleanValue
| functionCall # FunctionCallValue
;

```

أما القاعدة valueExpression بالرغم أنها تحتوي على قواعد و تفرعات أي label إلا أننا نحتاج لتخزين ال STRING ,NUMBER ,BOOLEAN أي من النوع التي ليس جميع تفرعاتها قواعد فستكون كلاس عادي و سترث من الكلاس AstNode.

٤- بناء ال visitor: كل اسم قاعدة أو كل label يجب أن يكون لها @override للتابع لزيارتها و تخزين البيانات ضمن الأوبجكت المنشأة من كلاسات الشجرة. اذا كان التابع باسم label ف رح يرجع من نوع الأب ويعمل ريتيرن ل visit اسم القاعدة النظامية و بتعمله casting للأب:

```
tag:
    standardTag # StandardTagElement
    | selfClosingTag # SelfClosingTagElement
    ;

standardTag :TAG_OPEN (H1| H2| H3| H4| H5| H6| STRONG|UL|P|DIV|LI|BUTTON|FORM|LABEL|INPUT) attribut
selfClosingTag : TAG_OPEN IMG attributes* TAG_SELF_CLOSE;
```

التابع يلي اسمه visitStanderTagElement ال return value من النوع Tag و فقط يعمل return (Tag)visitStanderTag لا يوجد فيه أي شيء آخر أما بالنسبة للتابع الذي يكون باسم القاعدة وليس ال label ال return value من نوعه هو و سيتم إنشاء أوبجكت من الكلاس الي باسم القاعدة و تخزين المعلومات المناسبة عن طريق التوابع المساعدة و عمل return للأوبجكت كما بالقاعدة SelfClosigTag , StanderTag.

٥- ال SymboleTable:

تم إنشاء كلاس اسمه Raw لتخزين ال position , line , value , type لأننا نحتاج إلى رقم السطر والعمود لمعرفة مكان الخطأ عند كشف الأخطاء ونحتاج إلى ال Type لتخزين النوع مثلاً ParameterSend , FunctionDec الخ..... ولدينا الكلاسات البقية لديه ليست من هذا الكلاس و setter,getter,print. وبال visitor اعمل اوبجكت من هاد ال symboleTable وخزن فيه المعلومات التي تفيدك بهذا الخطأ فقط وضعه بتابع هذه القاعدة و خزن المعلومات المناسبة عن طريق التوابع المساعدة.

## ٦- ايرور هاندلينغ:

يوجد لدينا كلاس اسمه Error يوجد فيه ليست من النصوص لتخزين الأخطاء بملف الأخطاء ويوجد فيه التوابع المساعدة وتابع print خاص بتخزين الأخطاء ضمن الملف. و يوجد لدينا كلاس semanticError لديه attributes لجميع ال symbolTables الموجودة بالإضافة إلى ال Error و التوابع المساعدة بالإضافة إلى function check لارجاع إذا كان يوجد أخطاء أو لا ويوجد أيضاً توابع لكل خطأ لمعالجته حيث يستقبل أوبجكت من ال symbolTable المناسب. بالشكل التالي:

سيتم شرح كل تابع وكيفية معالجة الخطأ بالصفحات القادمة إن شاء الله تعالى.

```
public boolean check() {...}

private boolean IncorrectAttribute(AttributeSymbolTable attributeSymbolTable) {...}

private boolean DetectCompositionError(DetectCompositionSymbolTable detectCompositionSymbolTable) {...}

private boolean DetectSelectorError(DetectSelectorSymbolTable detectSelectorSymbolTable) {...}

private boolean DetectTemplateError(DetectTemplateSymbolTable detectTemplateSymbolTable) {...}

private boolean RepeatedDeclarationFunction(DeclarationFunctionSymbolTable declarationFunctionSymbolTable) {...}

private boolean RepeatedDeclarationObjectInInterface(DeclarationObjectInInterfaceSymbolTable declarationObjectInInterfaceSymbolTable) {...}

private boolean IncorrectAttributeCss(AttributeCssSymbolTable attributeCssSymbolTable) {...}

private boolean ClassDeclarationAndStandalone(ClassDeclarationAndStandaloneSymbolTable classDeclarationAndStandaloneSymbolTable) {...}

private boolean Tags(TagsSymbolTable tagsSymbolTable) {...}
```

## ٧- code generation

بكل كلاس من كلاسات ال ast d يوجد @override لفنكشن الموجود بالكلاس AstNode الذي ستم تحويل الكود فيه واسمه generatedCode سيرجع سترينغ بشكل عودي من الاستدعاءات بحيث كلا كلاس يستدعي الأبناء و إرجاع الكود المحول إلى ال program و تخزينه ضمن الملفات js و ال html و ال css . سيتم شرح التحويل بالصفحات القادمة بإذن الله تعالى. وإلى الآن نكون قد انتهينا من بناء المترجم. تبقى شرح ال symbolTable وكيفية معالجة الأخطاء و سرد الأخطاء المعالجة وشرح قواعد التقابل والإضافات الأخرى.

خطوات عمل المشرع:

سيتم التنفيذ بدءاً من ال Main حيث سيتم عمل import لكافة المكاتب التي نحتاجها من antlr وبعد ذلك سنكتب التعليمات التالية لمسح الكود المصدر

```
CharStream st = fromFileName(source[i]);
angularLexer lexer = new angularLexer(st);
CommonTokenStream token = new CommonTokenStream(lexer);
angularParser parser = new angularParser(token);
ParseTree tree = parser.program();
Program doc = (Program) new AngularVisitor().visit(tree);
```

وعند عمل visit للشجرة بالتعليمة الأخيرة يتم عمل scan للبرنامج باستخدام ال Visitor pattern ويتم تخزين المعلومات اللازمة لكشف ال Semantic errors داخل ال Symbol table. لكل ملف تيسر سيتم إنشاء ملف لتخزين خرج ال ast بداخله وعند الكشف عن الأخطاء سيتم تخزينها بملف فقبل أن نخزن خرج ال code generation سيتم التأكد من أن الكود المصدري خال من الأخطاء وإلا لن يتم إنشاء ملفات الخرج سنبين ذلك بصورة إن شاء الله. ولتخزين الملفات يوجد التوابع التالية للتخزين

```
public static void generateJsFile(String js, String name) {...}

public static void generateCssFile(Program program, String fileName) {...}

public static void generateHtmlFile(String fileName, String fileNameOther) {...}

public static void astStorage(String ast, String name) {...}
```

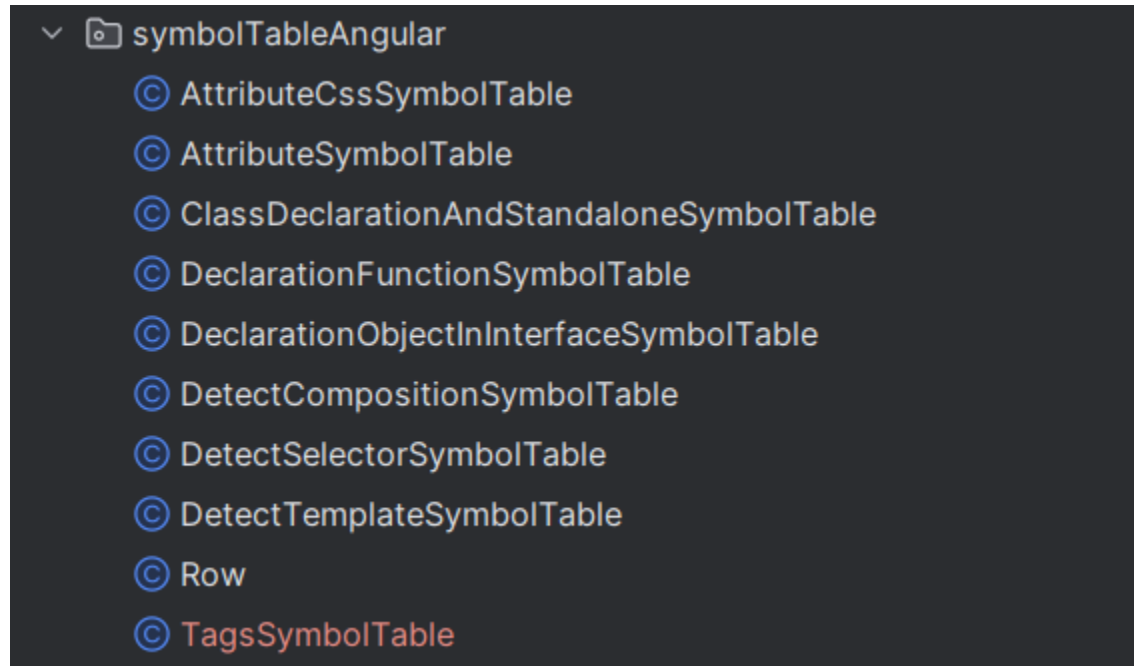
التابع الأول يخزن محتويات ملف ال js

الثاني يخزن محتويات ال css

الثالث يخزن محتويات ال html

الرابع يخزن محتويات ال ast وكل ملف سوف يسمى باسمه مع اللاحقة المناسبة.

## الـ symbolTables الموجودة:



جميعهم من نفس البنية التي ذكرناها سابقاً.

## كيفية معالجة الأخطاء:

سيتم استدعاء التابع check ضمن الـ visitor بالقاعدة الأولى حيث سيعمل على إسناد كل أوبجكت من الـ symbolTable للتابع المناسب لمعالجة الخطأ ضمنه وإذا كان يوجد أخطاء سيخزن بعدها نوع الخطأ بالشكل التالي:



```

public boolean check() { 1 usage  ASUS *
    boolean isValid = true;
    if (!IncorrectAttribute(this.attributeSymbolTable)) {
        error.getErrors().add("Exception Incorrect_Attribute");
        isValid = false;
    }
    if (!DetectCompositionError(this.detectCompositionSymbolTable)) {
        error.getErrors().add("Exception Composition Error");
        isValid = false;
    }
    if (!DetectSelectorError(this.detectSelectorSymbolTable)) {
        error.getErrors().add("Exception Selector Error");
        isValid = false;
    }
    if (!DetectTemplateError(this.detectTemplateSymbolTable)) {
        error.getErrors().add("Exception Template Error");
        isValid = false;
    }
    if (!RepeatedDeclarationFunction(this.declarationFunctionSymbolTable)) {
        error.getErrors().add("Exception RepeatedDeclarationFunction");
        isValid = false;
    }
    if (!RepeatedDeclarationObjectInInterface(this.declarationObjectInInterfaceSymbolTable)) {
        error.getErrors().add("Exception RepeatedDeclarationObjectInInterface");
        isValid = false;
    }
    if (!IncorrectAttributeCss(this.attributeCssSymbolTable)) {
        error.getErrors().add("Exception IncorrectAttributeCss");
        isValid = false;
    }
    if (!ClassDeclarationAndStandalone(this.classDeclarationAndStandaloneSymbolTable)) {
        error.getErrors().add("Exception IncorrectBodyComponent");
        isValid = false;
    }
    if (!Tags(this.tagsSymbolTable)) {
        error.getErrors().add("Exception Tage incorrect open and close");
        isValid = false;
    }
    return isValid;
}

```

التوابع التي عالجنا فيها الأخطاء:

```
private boolean IncorrectAttribute(AttributeSymbolTable attributeSymbolTable) {...}
private boolean DetectCompositionError(DetectCompositionSymbolTable detectCompositionSymbolTable) {...}
private boolean DetectSelectorError(DetectSelectorSymbolTable detectSelectorSymbolTable) {...}
private boolean DetectTemplateError(DetectTemplateSymbolTable detectTemplateSymbolTable) {...}
private boolean RepeatedDeclarationFunction(DeclarationFunctionSymbolTable declarationFunctionSymbolTable) {...}
private boolean RepeatedDeclarationObjectInInterface(DeclarationObjectInInterfaceSymbolTable declarationObjectInInterfaceSymbolTable) {...}
private boolean IncorrectAttributeCss(AttributeCssSymbolTable attributeCssSymbolTable) {...}
private boolean ClassDeclarationAndStandalone(ClassDeclarationAndStandaloneSymbolTable classDeclarationAndStandaloneSymbolTable) {...}
private boolean Tags(TagsSymbolTable tagsSymbolTable) {...}
```

التابع الأول لكشف الخطأ إن كان يوجد props لم يتم تعريفه ضمن ال interface وتم اسناد قيمة له ضمن التوابع أو عند تعريف مصفوفة من هذه ال interface و أيضاً اذا تكرر تعريف ال props أكثر من مرة اذا نسيت اسناد قيمة ل props موجودة بال interface أو اذا أسندت قيمة ل props غير موجودة بال interface أي سيعالج كافة الأنواع من هذا الخطأ ضمن نفس التابع. طريقة المعالجة بسيطة فقط من ال sympoletable سيتم مقارنة ال type اذا كان ضمن ال interface مع مقابلاتها ضمن تعريف المصفوفة و اسناد قيم و ليس أكثر من ذلك

التابع الثاني لكشف الخطأ detectedCompositionError وهو أنه عندما نعمل imports ضمن الكومبوننت لشيء لم يتم عمل import له بداية الكود أو إذا لم يتم عمل import ل component from Angular core

التابع الثالث لكشف الخطأ إذا لم يوجد selector او يوجد أكثر من selectore  
التابع الرابع لكشف الخطأ إذا لم يوجد Template او يوجد أكثر من Template  
التابع الخامس لكشف الخطأ إذا تم تعريف أكثر من تابع بنفس الاسم ونفس الباراميترات  
التابع السادس لكشف الخطأ إذا كررنا تعريف أو بجمكت ضمن ال interface  
التابع السابع لكشف الخطأ إذا تم إدخال attribute css غير موجودة بال css مثبتين بعض من ال css سنسردها ان شاء الله تعالى.

التابع الثامن لكشف الخطأ اذا كان ال standalone false و حاولنا تعريف كلاس  
التابع التاسع لكشف الأخطاء بالتاغات إذا فتحت تاغ ب h1 و سكرته ب h2 مثلاً

طريقة المعالجة سهلة و بسيطة اما ب list و اما ب map حسب نوع الخطأ.

سرد للأخطاء التي تمت معالجتها:

- 1-ايروور اذا بدى اعطي قيمة ل attribute غير موجودة بال interface
- 2-ايروور اذا عند اسناد قيم لل attribute و نسيت شي بدون اسناد
- 3-ايروور اذا عملت imports بقلب ال component لشي ماني عامله import باول الكود
- 4-ايروور اذا ما عمل import ل component من ال angular core
- 5-ايروور اذا ال template غير موجود
- 6-ايروور اذا selector غير موجود
- 7-ايروور تكرار ال function
- 8-ايروور تكرار اسم ال attribute عند تعريف ال interface
- 9-ايروور اذا كان ال standalone قيمته false ف ما بصير اعمل export class
- 10-ايروور اذا عم ضيف attribute css مانها موجودة بال css اساساً
- 11-ايروور اذا فتحت تاغ و سكرته بغير نوع

## شرح قواعد التقابل:

بعد الكشف عن الأخطاء و التأكد من عدم وجودها سيتم تجاهل ال import , imports , standalone

Selector,interface declaration

وبعدها سيتم التقابل على أساس القواعد التالية:

ال style سيأخذ مثل ما هو و تخزينه بملف ال css

أما بالنسبة لل template فان كل div أو form سيتم إنشاء function render له لعرض معلوماته على الواجهة.

محتويات ال render كل تاغ من نوع ال h1->h6 او p يحوي على سترينغات و ليس ts سيتم اسنادها ل innerHtml بالبداية ليتم عرضها مباشرة مثل

```
<h2 class="text-xl font-bold mb-4 text-center" style="color: #f5f7fa;">
```

```
this.carlist.innerHTML = `<h2 class = "text-xl font-bold mb-4 text-center"style = "color: #f5f7fa;">Cars </h2>`;
```

أما اذا كان ul و بداخلها li و يوجد \*ngFor سيتم تحويل المحتويات بالشكل التالي

```
<ul>
  <li *ngFor="let car of cars" class="car">
    <div class="car-info" (click)="selectCar(car)">
      <img [src]="car.car_image" alt="{{ car.car_name }}" />
      <p>{{ car.car_name }}</p>
    </div>
    <button class="delete-btn" (click)="deleteCar($event, car.id)">Delete</button>
    <button class="update-btn" (click)="updateCar($event, car.id)">Update</button>
  </li>
</ul>
```

```
this.carlist.innerHTML += this.cars.map(car => `
  <div class="car " data-id="${car.id}">
    <div class = "car-info" >
      <img src = "${car.car_image}" alt = "${ car.car_name }">
      <p >${car.car_name} </p>
    </div>
    <button class="delete-btn" >
      delete
    </button>
    <button class="update-btn" >
      Update
    </button>
  </div>
`).join('');
```

```

this.carlist.querySelectorAll('.car').forEach(item => {
  item.addEventListener('click', () => {
    this.selectCar(this.cars.find(p => p.id === parseInt(item.dataset.id)));
  });
  item.querySelector('.delete-btn').addEventListener('click', (e) => {
    this.deleteCar(parseInt(item.dataset.id), e);
  });
  item.querySelector('.update-btn').addEventListener('click', (e) => {
    this.updateCar(parseInt(item.dataset.id), e);
  });
});

```

أما بالنسبة لتعريف مصفوفة من ال interface

```

products: Product[] = [
  {id: 1, name: 'Product 1', image: '/images/1.jpg', details: 'Details about Product 1'},
  {id: 2, name: 'Product 2', image: '/images/2.jpg', details: 'Details about Product 2'},
  {id: 3, name: 'Product 3', image: '/images/3.jpg', details: 'Details about Product 3'}
];

```

سيتم تحويلها الى الشكل التالي مع اضافة نفس المصفوفة بعد ال || لم نضعها للمساحة فقط

```

const stored = localStorage.getItem('cars');
app.cars = JSON.parse(localStorage.getItem('cars')) ||

```

أما بالنسبة للتوابع فكل تعليمة لها مقابلها بالكود المقابل على الترتيب

```

selectCar(car: Car): void {
  this .selectedCar = car;
  car.sessionStorage.setItem('selectedCar');
  this.router.navigate('DetailsCarOnly');
}

```

```

selectCar(car) {
  this.selectedCar = car;
  sessionStorage.setItem( 'selectedCar', JSON.stringify(car));
  window.location.href = 'DetailsCarOnly.html';
}

```

```

deleteCar( carId: number,event: MouseEvent): void {
    event.stopPropagation();
    if (confirm('Are you sure to delete this car from list ?')){
        this .cars = this .cars . filter (car => car . id != carId);
        cars.localStorage.setItem('cars');
        if ('this.selectedCar && this.selectedCar.id === carId') {
            this .selectedCar = null;
        }
        sessionStorage.clear();
        this.render();
    }
    else
    {
        this.render();
    }
}

```

```

deleteCar(carId,event) {
    event.stopPropagation();
    if (confirm('Are you sure to delete this car from list ?')) {
        this.cars = this.cars.filter(car => car.id !== carId);
        localStorage.setItem('cars', JSON.stringify(this.cars));
        if (this.selectedCar && this.selectedCar.id === carId) {
            this.selectedCar = null;
        }
        sessionStorage.clear();
        this.render();
    }
    else {
        this.render();
    }
}

```

```

updateCar( carId: number,event: MouseEvent): void {
    event.stopPropagation();
    if (confirm('Are you sure to update this car ?')){
        carId.sessionStorage.setItem('selectedCarToUpdate');
        this.router.navigate('UpdateCar');
    }
    this.render();
}
}

```

```

updateCar(carId,event) {
    event.stopPropagation();
    if (confirm('Are you sure to update this car ?')) {
        sessionStorage.setItem( 'selectedCarToUpdate', JSON.stringify(carId));
        window.location.href = 'UpdateCar.html';
        this.render();
    }
}

```

أما بالنسبة لترتيب ال template ف يوجد لدينا تابع بكل ملف js بحيث يضع الستر كشر الخاص بالصفحة

```

createAppStructure() {
    this.appContainer = document.getElementById('app');
    const header = document.createElement('div');
    header.innerHTML = ``;
    this.appContainer.appendChild(header);
    this.container = document.createElement('div');
    this.container.className = 'container';
    this.appContainer.appendChild(this.container);
    this.carlist = document.createElement('div');
    this.carlist.className = 'car-list';
    this.container.appendChild(this.carlist);
}

```

أما بالنسبة لبداية ملف ال js يجب أن يبدأ ب

```
document.addEventListener('DOMContentLoaded', () => {  
  const app = new AppComponent();  
  app.createAppStructure();  
  app.render();  
});
```

و إذا كان يوجد تعريف لمصفوفة من اللوكال ستوريج أو متغير ما سيتم وضعه هنا قبل عملية ال render

هكذا أصبح لدينا جزء من قواعد التقابل و ترتيب ملف الخرج بال js على الشكل التالي  
كلاس يوجد بداخله تابع الستركشر و توابع ال render و التوابع الموجودة وتابع render يستدعي بداخله  
جميع توابع ال render الأخرى.  
و بعدها ال document.addEventListener بداخله أوبجكت من هذا الكلاس و استدعاء تابع الستركشر  
و بعدها التخزين و من ثم استدعاء تابع ال render.

سيصبح الكود النهائي بالشكل التالي:

```
class AppComponent {  
  createAppStructure() {  
    this.appContainer = document.getElementById('app');  
    const header = document.createElement('div');  
    header.innerHTML = ``;  
    this.appContainer.appendChild(header);  
    this.container = document.createElement('div');  
    this.container.className = 'container';  
    this.appContainer.appendChild(this.container);  
    this.carlist = document.createElement('div');  
    this.carlist.className = 'car-list';  
    this.container.appendChild(this.carlist);  
  }  
}
```



```

rendercarlist() {
  this.carlist.innerHTML = `<h2 class = "text-xl font-bold mb-4 text-center" style = "color: #f5f7fa;">Cars </h2>`;

  if (this.cars.length === 0) {
    this.carlist.innerHTML += `
    <h3 class="mb-4 empty-message">No cars available<br>Add cars to see them listed here</h3>`;
    return;
  }

  this.carlist.innerHTML += this.cars.map(car => `
  <div class="car " data-id="${car.id}">
  <div class = "car-info" >
  <img src = "${car.car_image}" alt = "${car.car_name}">
  <p >${car.car_name} </p>
  </div>
  <button class="delete-btn" >
  delete
  </button>
  <button class="update-btn" >
  Update
  </button>
  </div>
  `).join('');

  this.carlist.querySelectorAll('.car').forEach(item => {
    item.addEventListener('click', () => {
      this.selectCar(this.cars.find(p => p.id === parseInt(item.dataset.id)));
    });
    item.querySelector('.delete-btn').addEventListener('click', (e) => {
      this.deleteCar(parseInt(item.dataset.id), e);
    });
    item.querySelector('.update-btn').addEventListener('click', (e) => {
      this.updateCar(parseInt(item.dataset.id), e);
    });
  });
}

render() {
  this.rendercarlist()
}

```

```

selectCar(car) {
  this.selectedCar = car;
  sessionStorage.setItem( 'selectedCar', JSON.stringify(car));
  window.location.href = 'DetailsCarOnly.html';
}

deleteCar(carId,event) {
  event.stopPropagation();
  if (confirm('Are you sure to delete this car from list ?')) {
    this.cars = this.cars.filter(car => car.id !== carId);
    localStorage.setItem('cars', JSON.stringify(this.cars));
    if (this.selectedCar && this.selectedCar.id === carId) {
      this.selectedCar = null;
    }
    sessionStorage.clear();
    this.render();
  }
  else {
    this.render();
  }
}

updateCar(carId,event) {
  event.stopPropagation();
  if (confirm('Are you sure to update this car ?')) {
    sessionStorage.setItem( 'selectedCarToUpdate', JSON.stringify(carId));
    window.location.href = 'UpdateCar.html';
    this.render();
  }
}

```

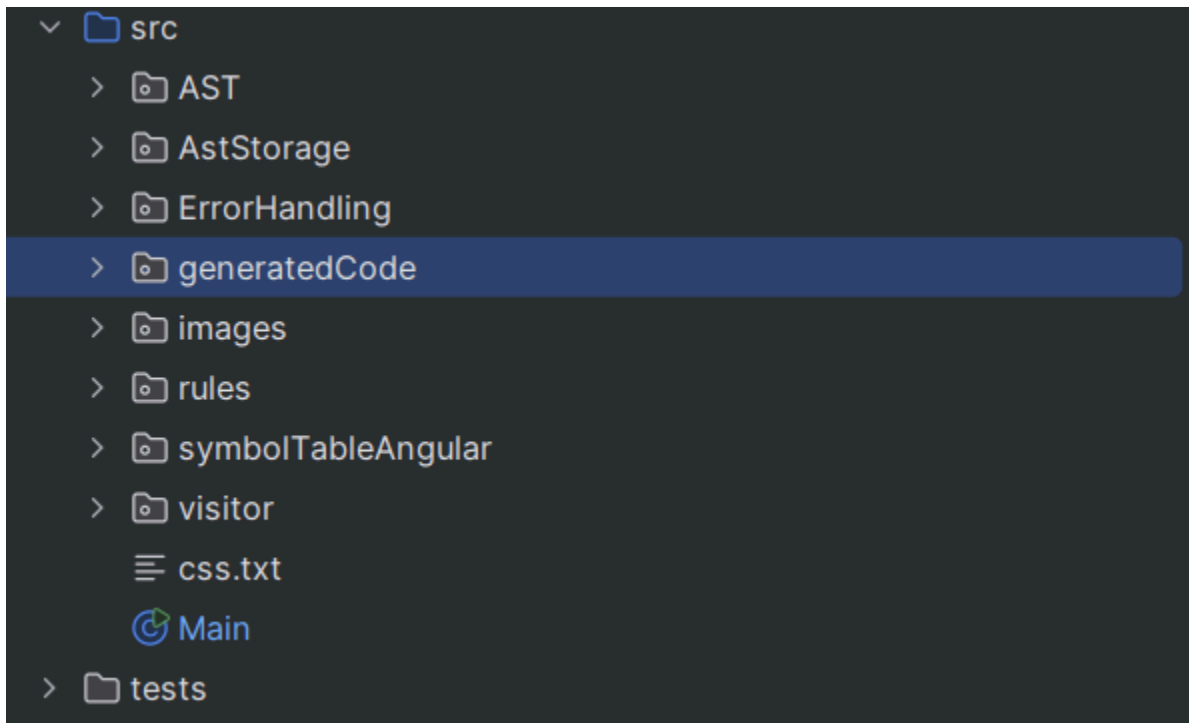
```

document.addEventListener('DOMContentLoaded', () => {
  const app = new AppComponent();
  app.createAppStructure();
  const stored = localStorage.getItem('cars');
  app.cars = JSON.parse(localStorage.getItem('cars')) || [
    {id: 1,car_name: "MS",car_image: "/images/8.jpg",car_price: 4444000,car_brand: "BMW",car_model: "MS series",car_year: 2020,car_category: "Sport",car_color: "Black",car_engine_size: "2.5L",car_engine_type: "Gasoline",car_},
    {id: 2,car_name: "G class",car_image: "/images/12.jpg",car_price: 59544,car_brand: "Mercedes",car_model: "G class series",car_year: 2018,car_category: "Jeep",car_color: "Purple",car_engine_size: "3L",car_engine_type: "D"},
    {id: 3,car_name: "Land Cruiser",car_image: "/images/7.jpg",car_price: 25944,car_brand: "Toyota",car_model: "Land Cruiser series",car_year: 2022,car_category: "Jeep",car_color: "White",car_engine_size: "2L",car_engine_ty},
    {id: 4,car_name: "Lexus TX",car_image: "/images/9.jpg",car_price: 345435,car_brand: "Lexus",car_model: "TX series",car_year: 2019,car_category: "Jeep",car_color: "White",car_engine_size: "2.5L",car_engine_type: "Electri},
    {id: 5,car_name: "GLE",car_image: "/images/10.jpg",car_price: 4444000,car_brand: "Mercedes",car_model: "GLE series",car_year: 2023,car_category: "SUV",car_color: "Black",car_engine_size: "3.5L",car_engine_type: "Gasolin},
  ];
  localStorage.setItem('cars', JSON.stringify(app.cars));
  app.render();
});

```

ونفس الأمر كذلك بالنسبة لل form وباقي القواعد.  
كل قاعدة بملف المصدر لها القاعدة المقابلة لها بملف ال js .  
من الصعب جداً كتابة جميعها هنا.  
بالإمكان تشغيل الكود من الغيت هب وعمل رن له والنظر بالقواعد المعالجة.

تقسيمات المشروع:

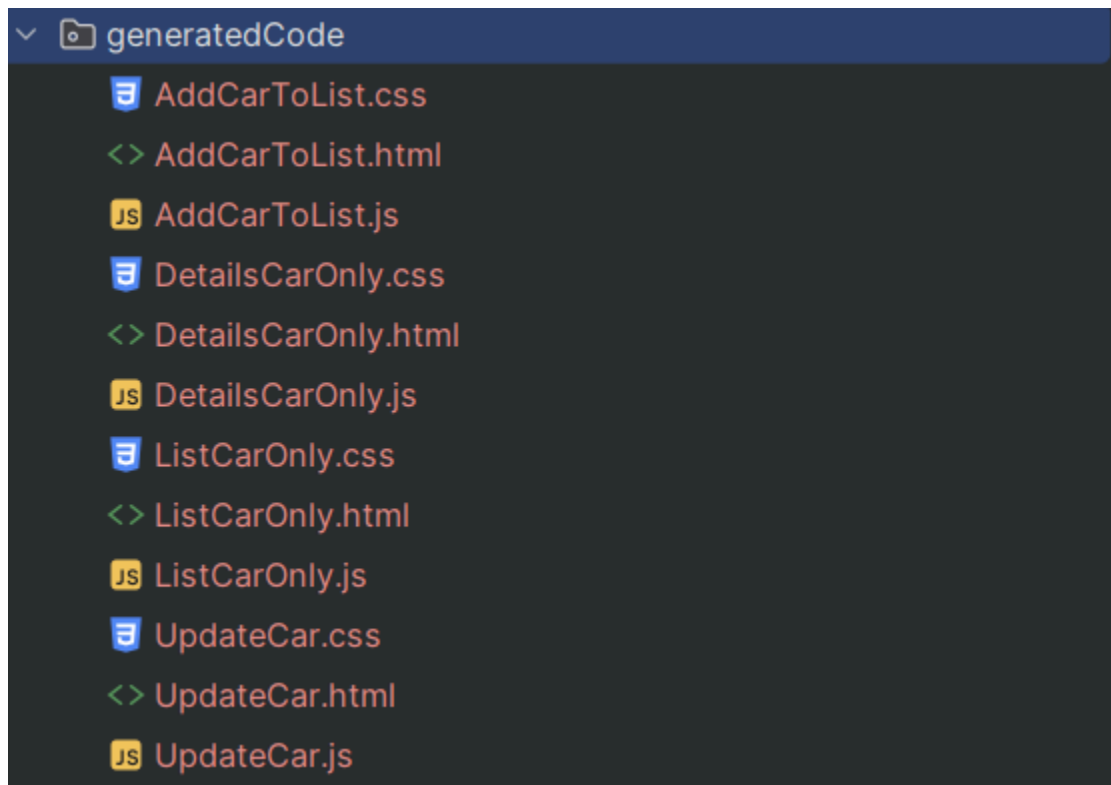


مجلد ال ast يحتوي على كلاسات الشجرة  
مجلد ال astStorage يتم تخزين خرج ال ast  
مجلد ال ErrorHandler يحوي على كلاسات ال Error مع ملف لتخزين ال errors  
النتيجة .

مجلد ال generatedCode يحوي على ملفات الخرج بعد التحويل  
مجلد ال images يحوي على بعض الصور للتجريب.  
مجلد ال rules يحوي على ال lexer و ال parser  
مجلد ال symbolTableAngular يحوي على كلاسات ال symbolTable

مجلد ال visitor يحتوي على الvisitore  
مجلد ال test يحتوي على ملفات الكود المصدري أنغيولار





ال css التي تم تحقيقها لعدم إظهار خطأ و بالامكان الإضافة بأي وقت

```
List<String> CssAttr = Arrays.asList(
    "display", "width", "gap", "border-right", "list-style-type", "padding", "align-items", "border-bottom",
    "cursor", "height", "object-fit", "margin-bottom", "margin", "min-height", "padding-right", "background",
    "border-radius", "flex", "flex-direction", "border", "max-width", "font-size", "box-shadow", "text-align",
    "margin-top", "color", "justify-content", "direction", "overflow", "font-weight", "unicode-bidi", "outline",
    "resize", "padding-left", "padding-right", "border-color");
```

بعض الأشياء الإضافية :

تخزين خرج الشجرة بملف

تخزين الأخطاء بملف

طباعة على الكونسول للملفات التي تم إنشائها أو لا حسب إذا موجود ملفات أو لا

على الشكل التالي:

```
"C:\Program Files\Java\jdk1.8.0_291\bin\java.exe" ...
The Files ListCarOnly.(js,css,html) Generated Successfully! and no Errors in code
The Files DetailsCarOnly.(js,css,html) Generated Successfully! and no Errors in code
The Files AddCarToList.(js,css,html) Generated Successfully! and no Errors in code
The Files UpdateCar.(js,css,html) Generated Successfully! and no Errors in code
```

```
The Files ListCarOnly.(js,css,html) Generated Successfully! and no Errors in code
The Files DetailsCarOnly.(js,css,html) Generated Successfully! and no Errors in code
The Files AddCarToList.(js,css,html) Generated Successfully! and no Errors in code
The Files UpdateCar.(js,css,html) Doesn't Generated Because their are Errors
```

ال error التي تمت معالجتها :

```
At Line 58 in position 102 Attribute price not found in interface
At Line 59 in position 6 Attribute id not found in Assignment
Exception Incorrect_Attribute
At Line 13 in position 2 imports CommonModule not found in import statement
import Component not found in import statement
Exception Composition Error
template not found in Component Declaration
selector not found in Component Declaration
Exception Selector|Template Error
At Line 69 in position 7 Function was Decelerated already
Exception RepeatedDeclarationFunction
At Line 4 in position 2 Object was Decelerated already
Exception RepeatedDeclarationObjectInInterface
At Line 17 in position 13 Css attribute not found in Body of Css
Exception IncorrectAttributeCss
At Line 56 in position 0 classDeclaration Error the standalone is false or not found
Exception IncorrectBodyComponent
```

```
At Line 115 in position 2 Tags Error the h1 tag not close in the same tage
Tags Error the label tag not open in the same tage
Exception Tage incorrect open and close
```











خرج ال error عندما يكون أكثر من ملف مصدري حيث يكتب اسم الملف و الأخطاء تحته ثم الملف الثاني وهكذا

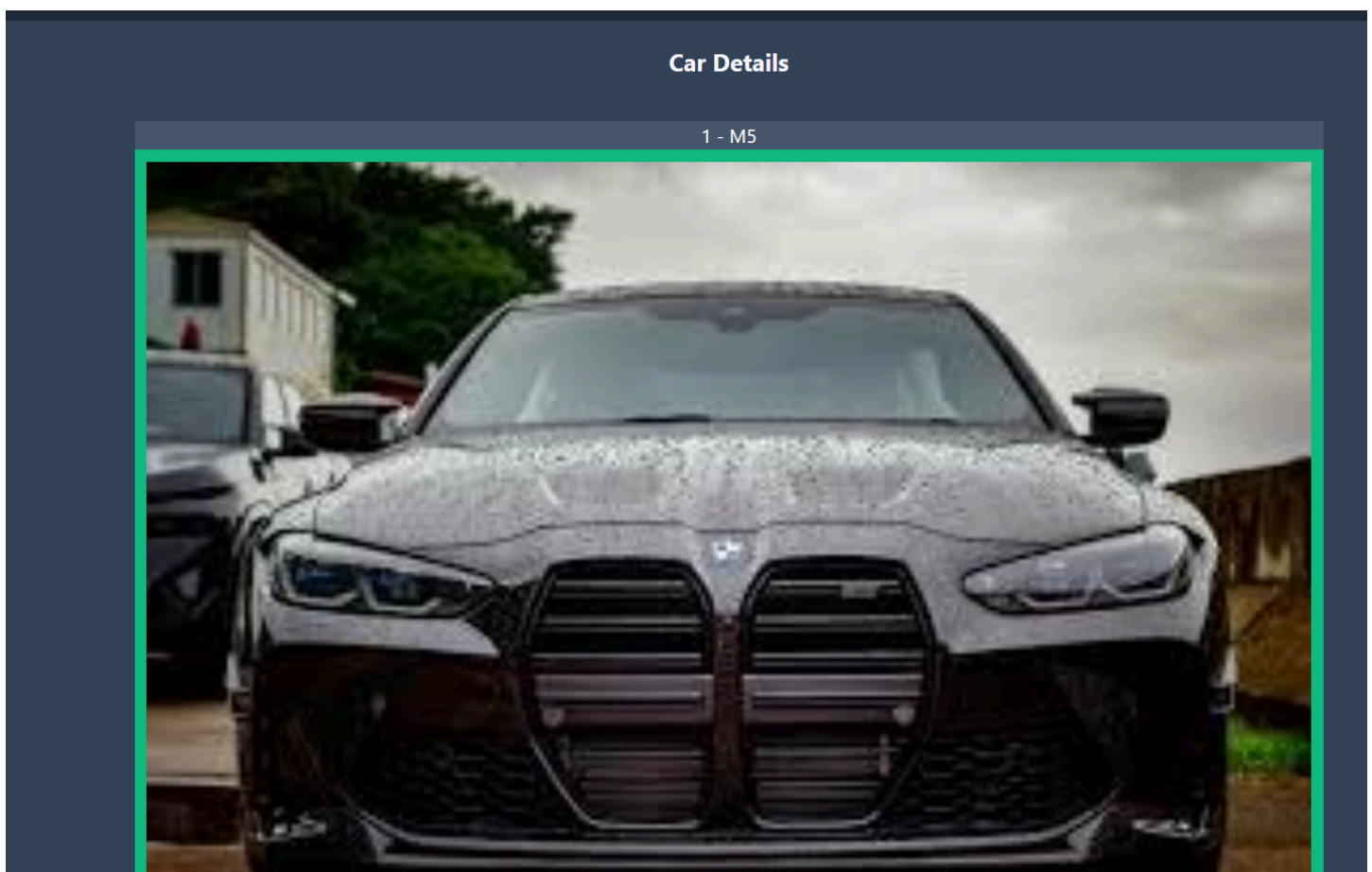
```
ListCarOnly
No Error in this test
DetailsCarOnly
No Error in this test
AddCarToList
No Error in this test
UpdateCar
At Line 115 in position 2 Tags Error the h1 tag not close in the same tage
Tags Error the label tag not open in the same tage
Exception Tage incorrect open and close
|
```





بعض الخرج من الواجهات :

Cars			<a href="#">Go to AddCarToIstPage</a>
	M5	 <a href="#">Update</a>	
	G class	 <a href="#">Update</a>	
	Land Cruiser	 <a href="#">Update</a>	
	Lexus TX	 <a href="#">Update</a>	
	GLE	 <a href="#">Update</a>	





**car\_name**

Enter car\_name

**car\_image**

Choose File No file chosen

**car\_price**

Enter car\_price

**car\_brand**

Enter car\_brand

**car\_model**

Enter car\_model

**car\_year**

Enter car\_year

انتهى التقرير