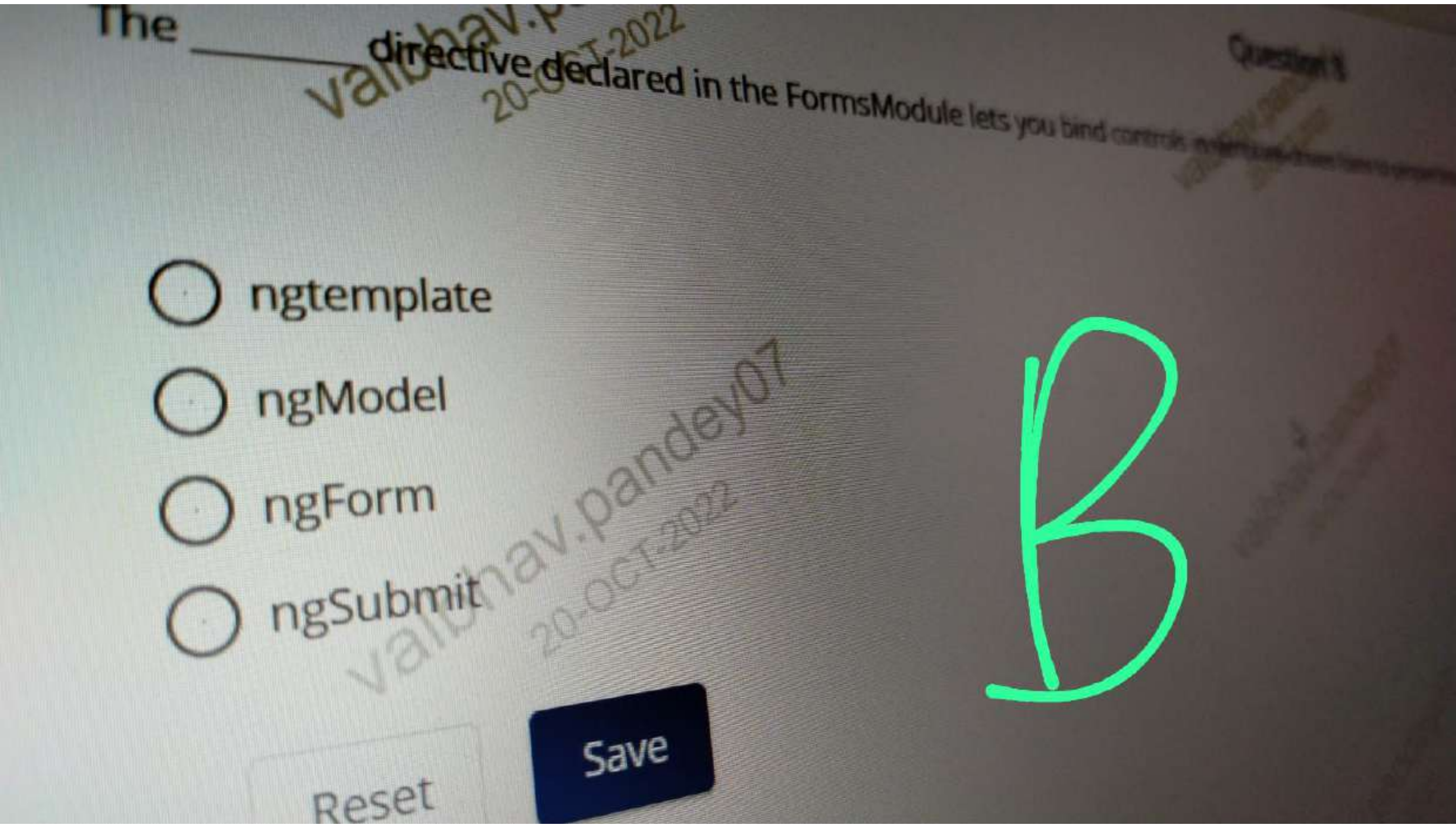


Question 8

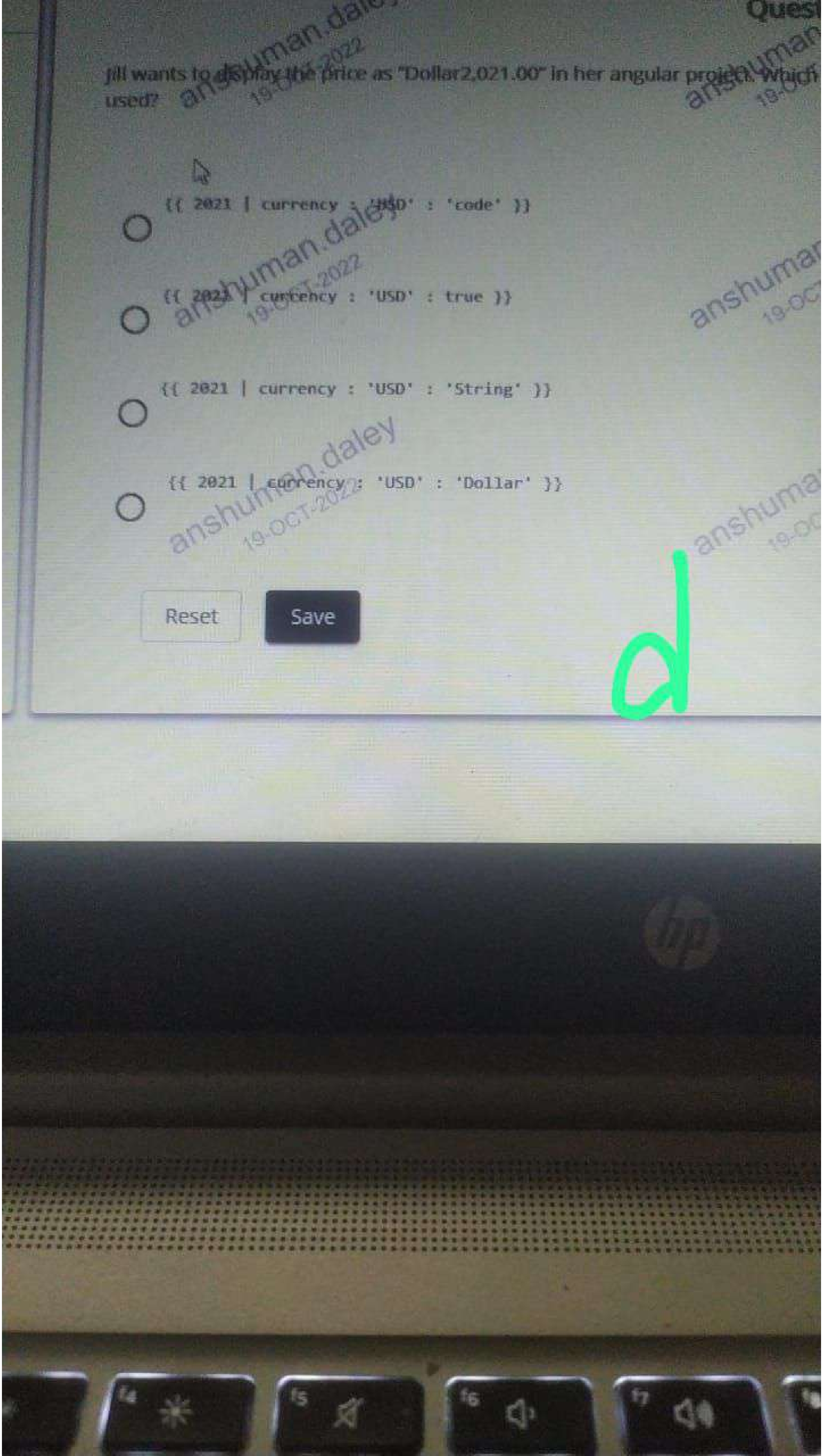
The ________ directive declared in the FormsModule lets you bind controls [illegible]

○ ngtemplate

○ ngModel

○ ngForm

○ ngSubmit



Quest

Jill wants to display the price as "Dollar2,021.00" in her angular project. Which
used?

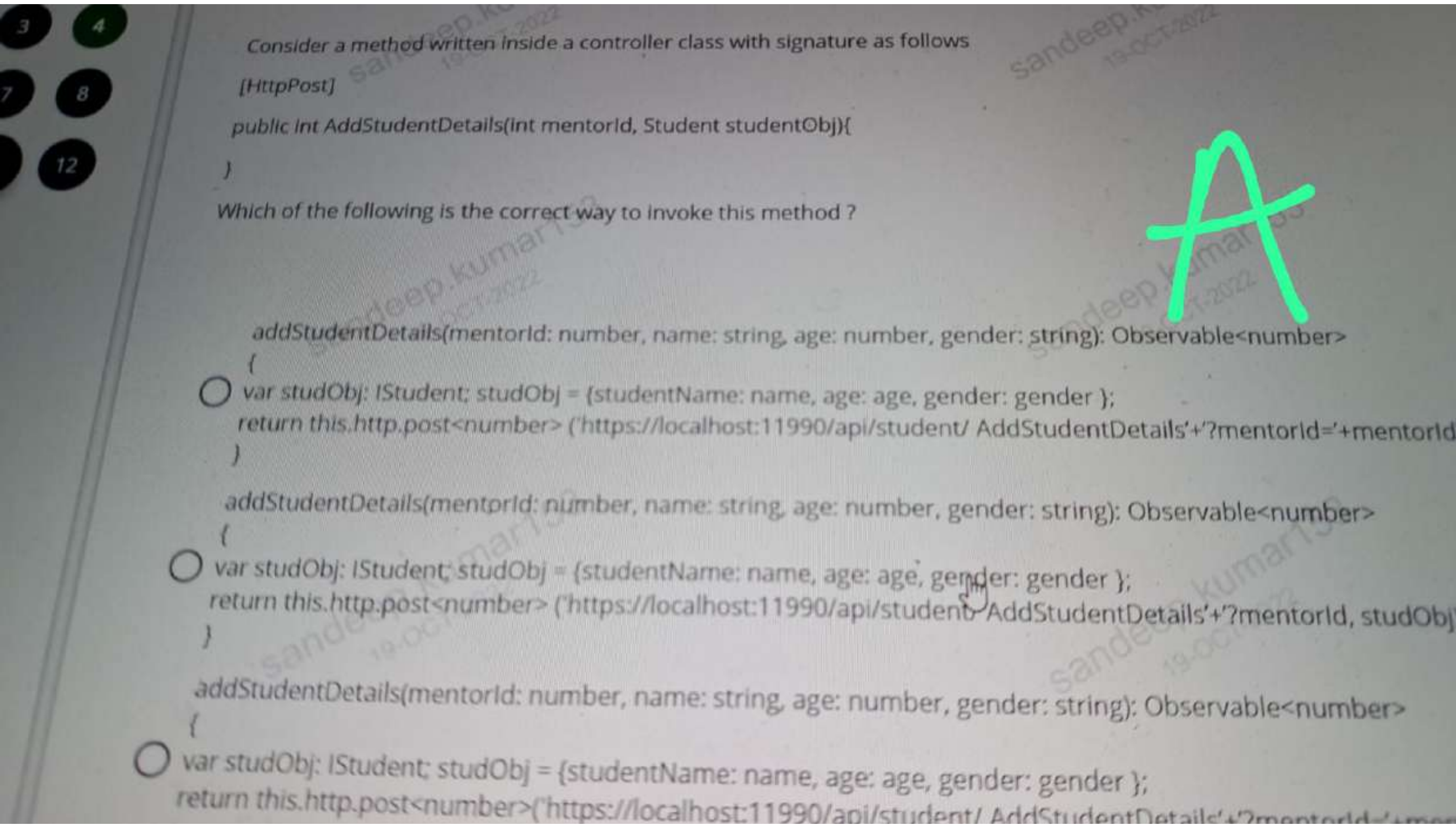
○ `{{ 2021 | currency : 'USD' : 'code' }}`

○ `{{ 2021 | currency : 'USD' : true }}`

○ `{{ 2021 | currency : 'USD' : 'String' }}`

○ `{{ 2021 | currency : 'USD' : 'Dollar' }}`

d



Consider a method written inside a controller class with signature as follows

```
[HttpPost]
public int AddStudentDetails(int mentorId, Student studentObj){
}
```

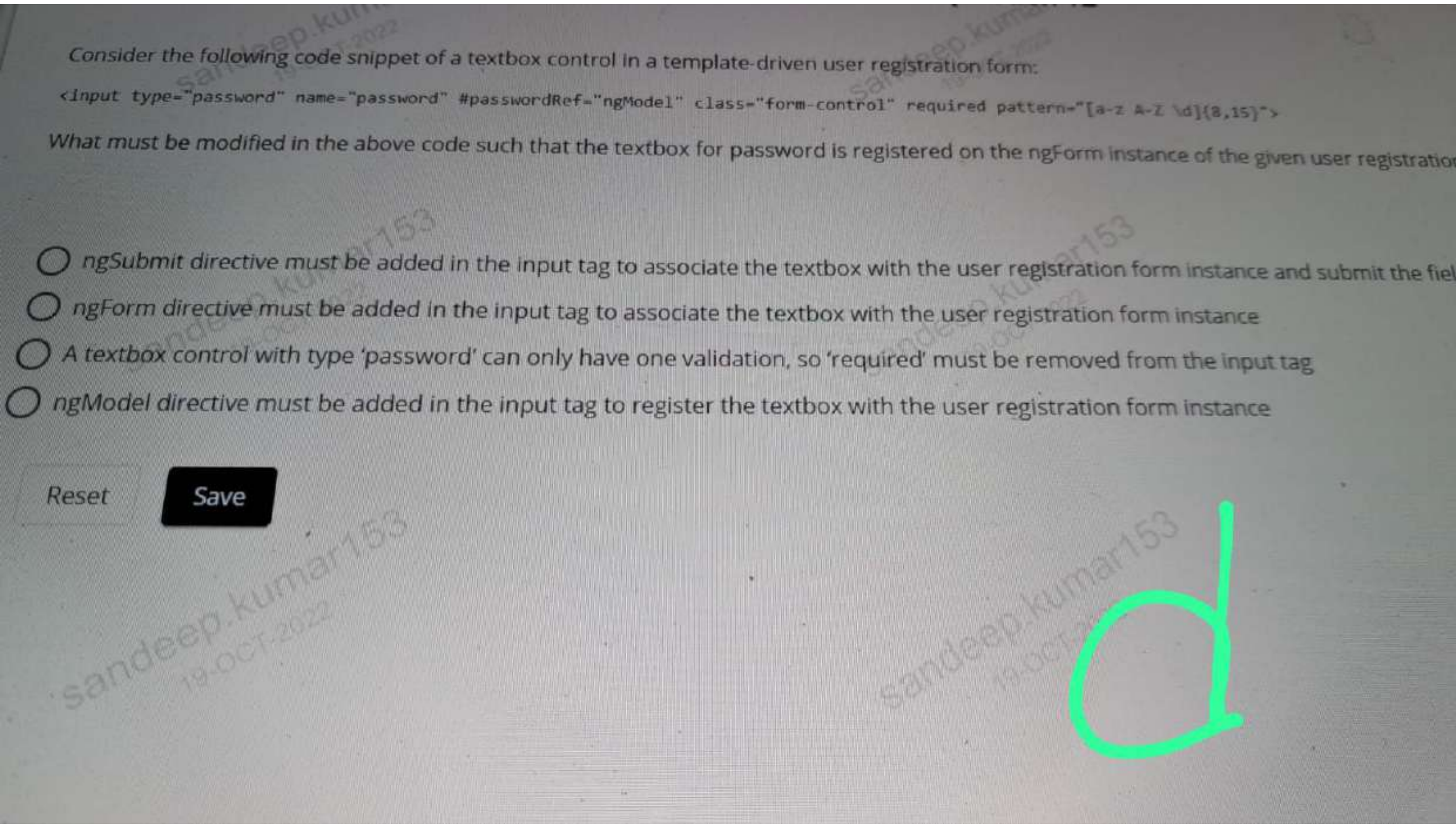
Which of the following is the correct way to invoke this method ?

A

```
  addStudentDetails(mentorId: number, name: string, age: number, gender: string): Observable<number>
  {
○ var studObj: IStudent; studObj = {studentName: name, age: age, gender: gender };
  return this.http.post<number> ('https://localhost:11990/api/student/ AddStudentDetails'+'?mentorId='+mentorId
  }
```

```
  addStudentDetails(mentorId: number, name: string, age: number, gender: string): Observable<number>
  {
○ var studObj: IStudent; studObj = {studentName: name, age: age, gender: gender };
  return this.http.post<number> ('https://localhost:11990/api/student/AddStudentDetails'+'?mentorId, studObj
  }
```

```
  addStudentDetails(mentorId: number, name: string, age: number, gender: string): Observable<number>
  {
○ var studObj: IStudent; studObj = {studentName: name, age: age, gender: gender };
  return this.http.post<number>('https://localhost:11990/api/student/ AddStudentDetails'+'?mentorId='+men
```



Consider the following code snippet of a textbox control in a template-driven user registration form:

```
<input type="password" name="password" #passwordRef="ngModel" class="form-control" required pattern="[a-z A-Z \d]{8,15}">
```

What must be modified in the above code such that the textbox for password is registered on the ngForm instance of the given user registratio

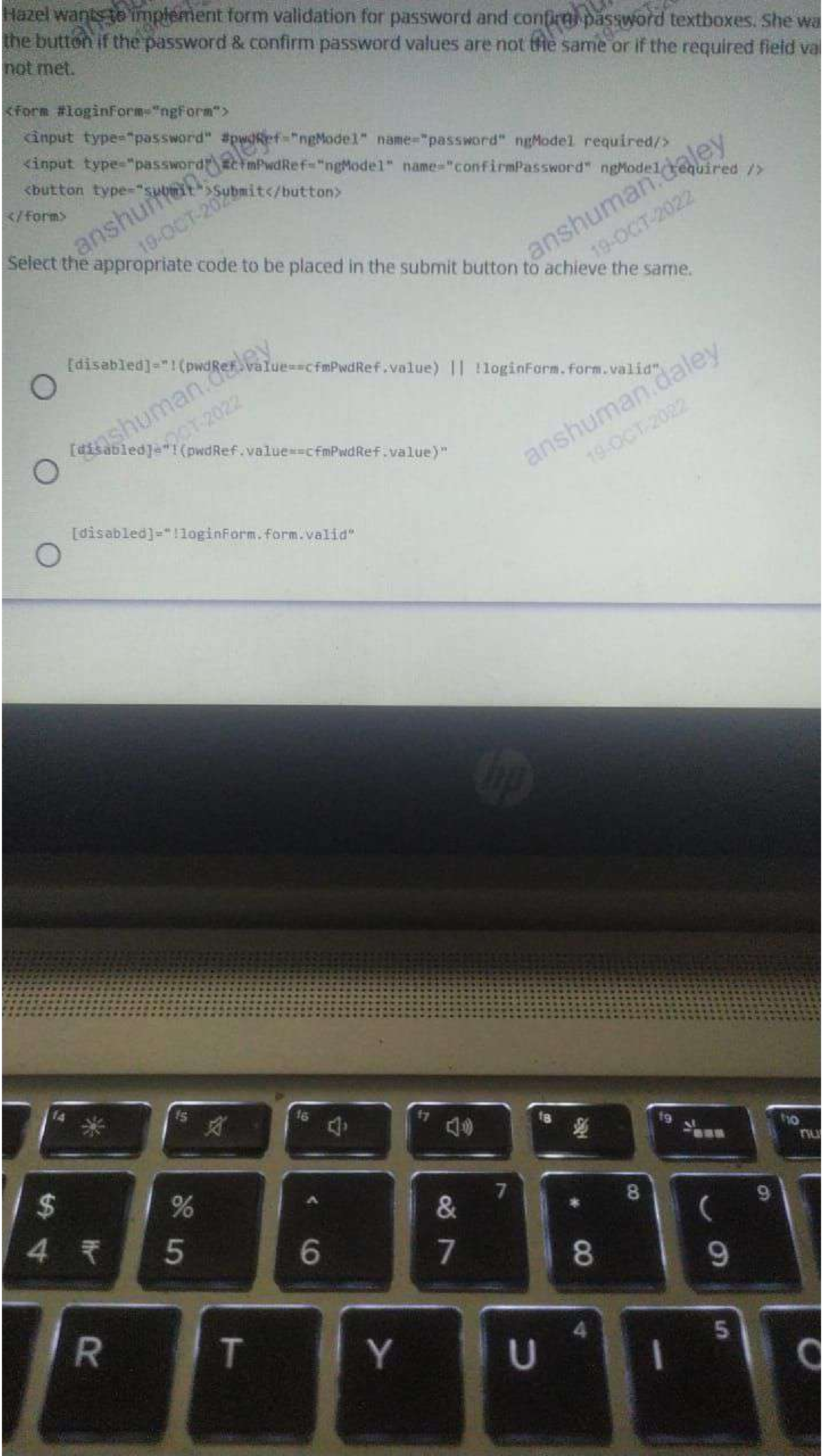
○ ngSubmit directive must be added in the input tag to associate the textbox with the user registration form instance and submit the fiel

○ ngForm directive must be added in the input tag to associate the textbox with the user registration form instance

○ A textbox control with type 'password' can only have one validation, so 'required' must be removed from the input tag

○ ngModel directive must be added in the input tag to register the textbox with the user registration form instance

d



Hazel wants to implement form validation for password and confirm password textboxes. She wa
the button if the password & confirm password values are not the same or if the required field va
not met.

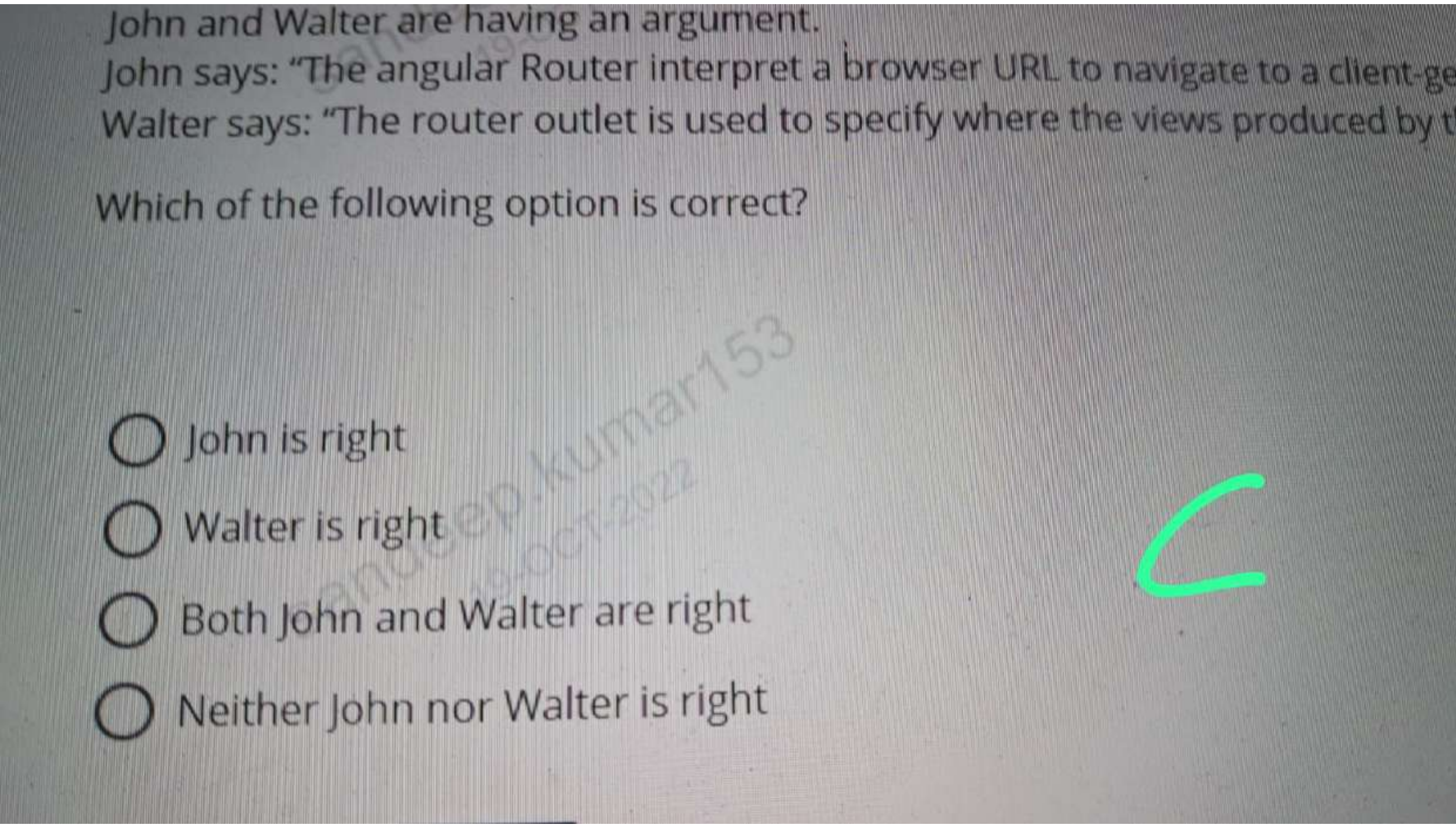
```
<form #loginForm="ngForm">
  <input type="password" #pwdRef="ngModel" name="password" ngModel required/>
  <input type="password" #cfmPwdRef="ngModel" name="confirmPassword" ngModel required />
  <button type="submit">Submit</button>
</form>
```

Select the appropriate code to be placed in the submit button to achieve the same.

○ `[disabled]="!(pwdRef.value==cfmPwdRef.value) || !loginForm.form.valid"`

○ `[disabled]="!(pwdRef.value==cfmPwdRef.value)"`

○ `[disabled]="!loginForm.form.valid"`



John and Walter are having an argument.
John says: "The angular Router interpret a browser URL to navigate to a client-ge
Walter says: "The router outlet is used to specify where the views produced by t

Which of the following option is correct?

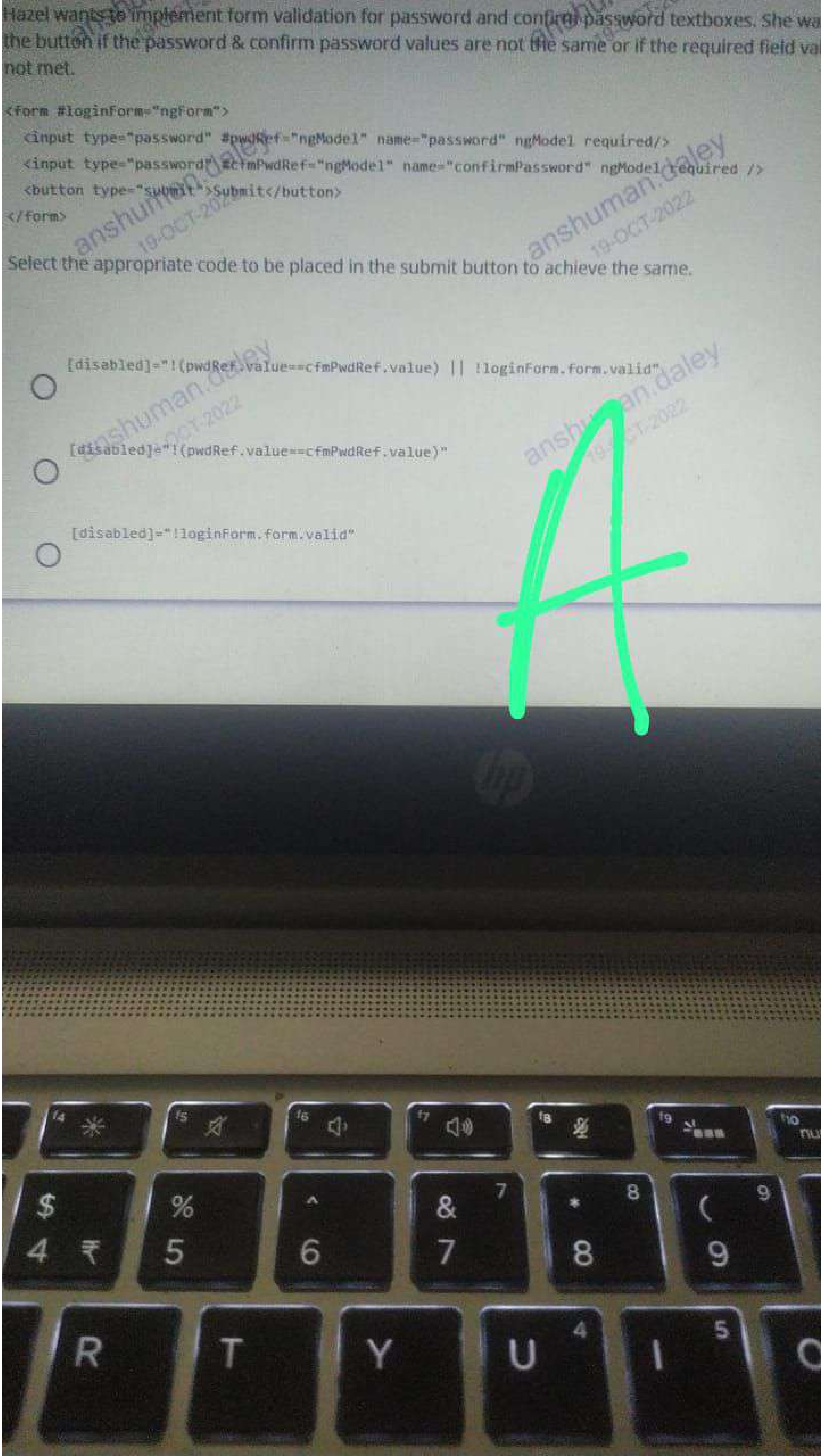
○ John is right

○ Walter is right

○ Both John and Walter are right

○ Neither John nor Walter is right

C



Hazel wants to implement form validation for password and confirm password textboxes. She wa
the button if the password & confirm password values are not the same or if the required field va
not met.

```
<form #loginForm="ngForm">
  <input type="password" #pwdRef="ngModel" name="password" ngModel required/>
  <input type="password" #cfmPwdRef="ngModel" name="confirmPassword" ngModel required />
  <button type="submit">Submit</button>
</form>
```

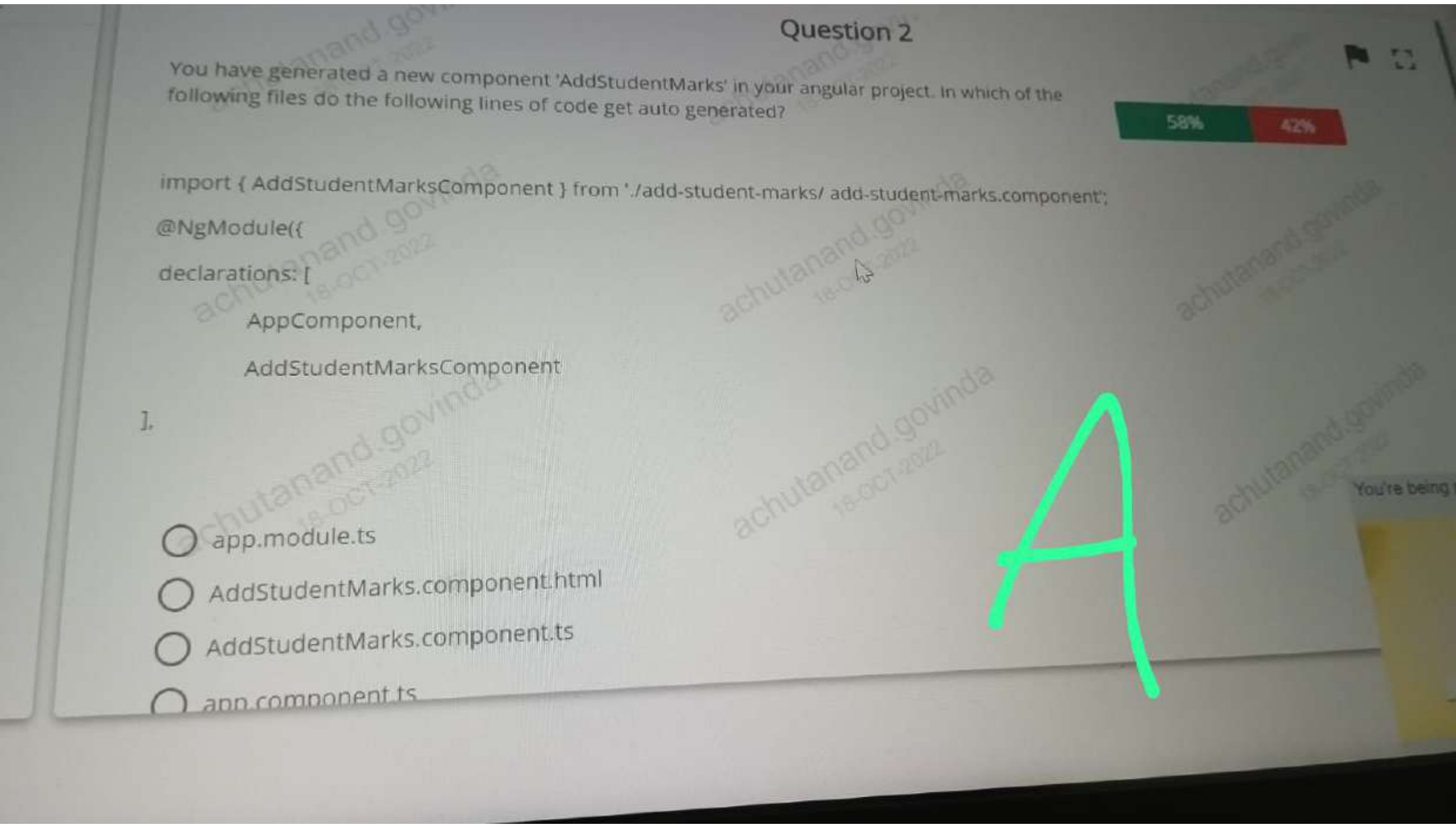
Select the appropriate code to be placed in the submit button to achieve the same.

○ `[disabled]="!(pwdRef.value==cfmPwdRef.value) || !loginForm.form.valid"`

○ `[disabled]="!(pwdRef.value==cfmPwdRef.value)"`

○ `[disabled]="!loginForm.form.valid"`

A



Question 2

You have generated a new component 'AddStudentMarks' in your angular project. In which of the following files do the following lines of code get auto generated?

```
import { AddStudentMarksComponent } from './add-student-marks/ add-student-marks.component';
@NgModule({
declarations: [
        AppComponent,
        AddStudentMarksComponent
],
```

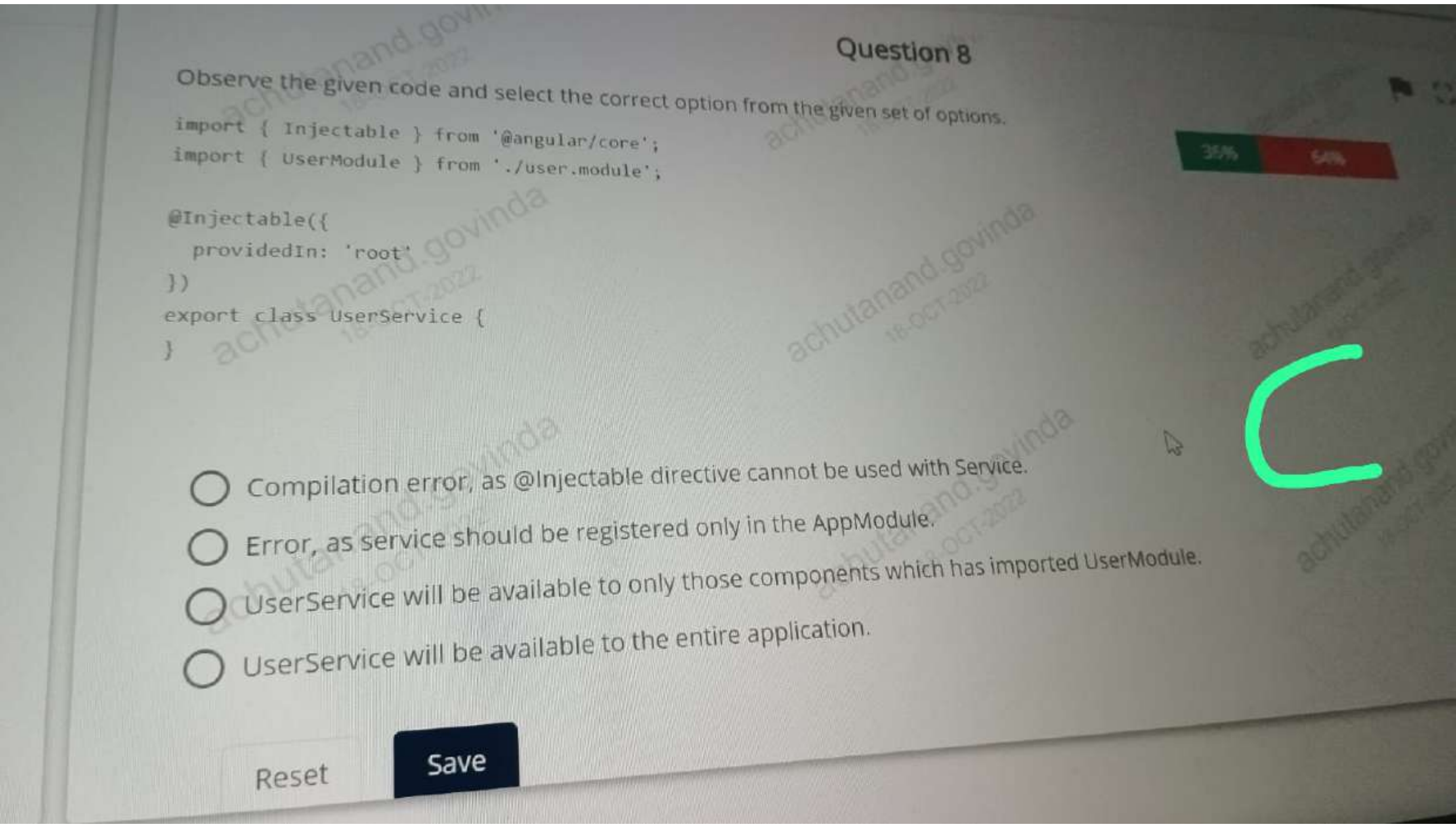
○ app.module.ts

○ AddStudentMarks.component.html

○ AddStudentMarks.component.ts

○ app.component.ts

A



Question 8

Observe the given code and select the correct option from the given set of options.

```
import { Injectable } from '@angular/core';
import { UserModule } from './user.module';

@Injectable({
  providedIn: 'root'
})
export class UserService {
}
```

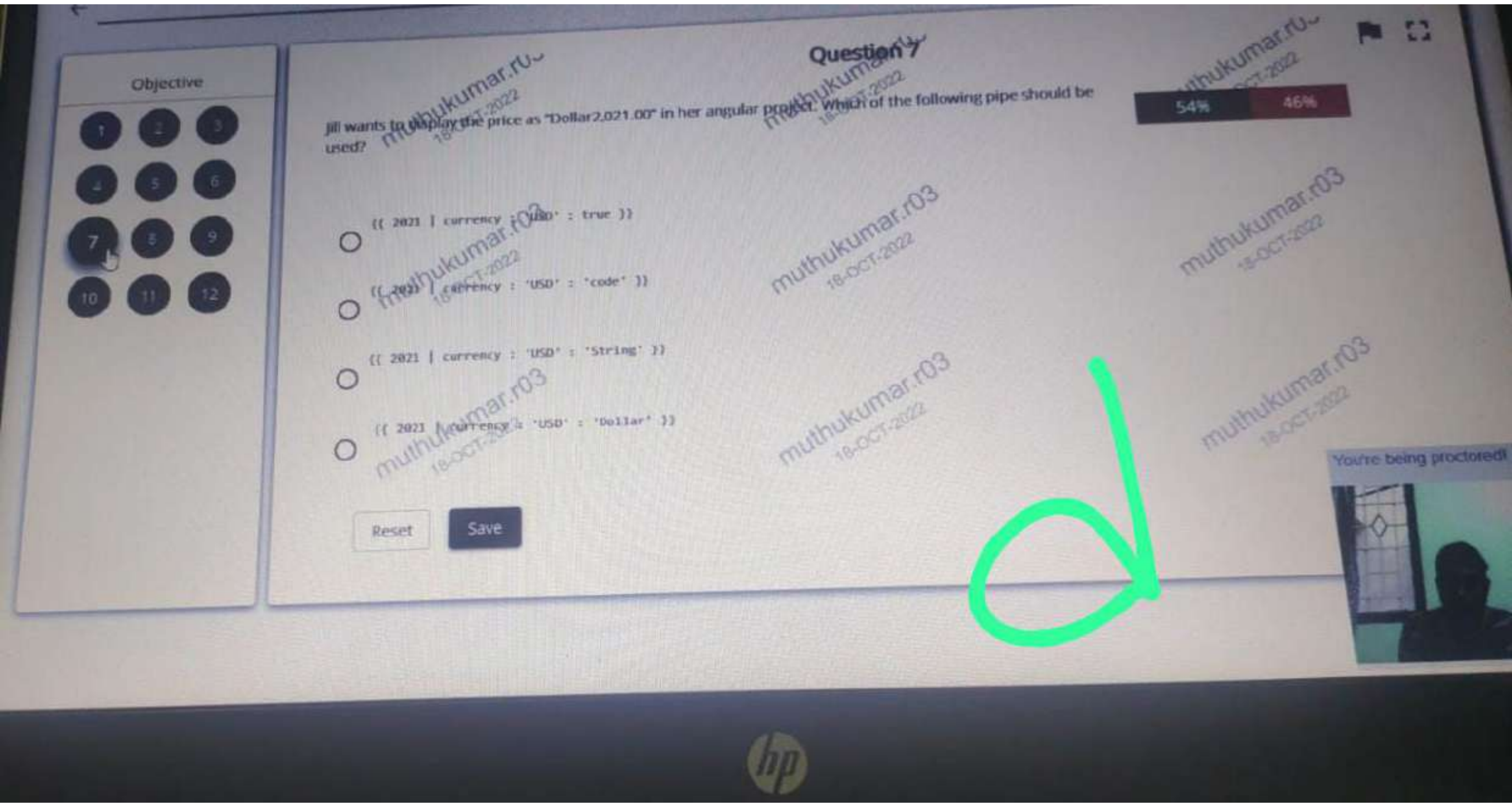
○ Compilation error, as @Injectable directive cannot be used with Service.

○ Error, as service should be registered only in the AppModule.

○ UserService will be available to only those components which has imported UserModule.

○ UserService will be available to the entire application.

C



Question 7

Jill wants to display the price as "Dollar2,021.00" in her angular project. Which of the following pipe should be used?

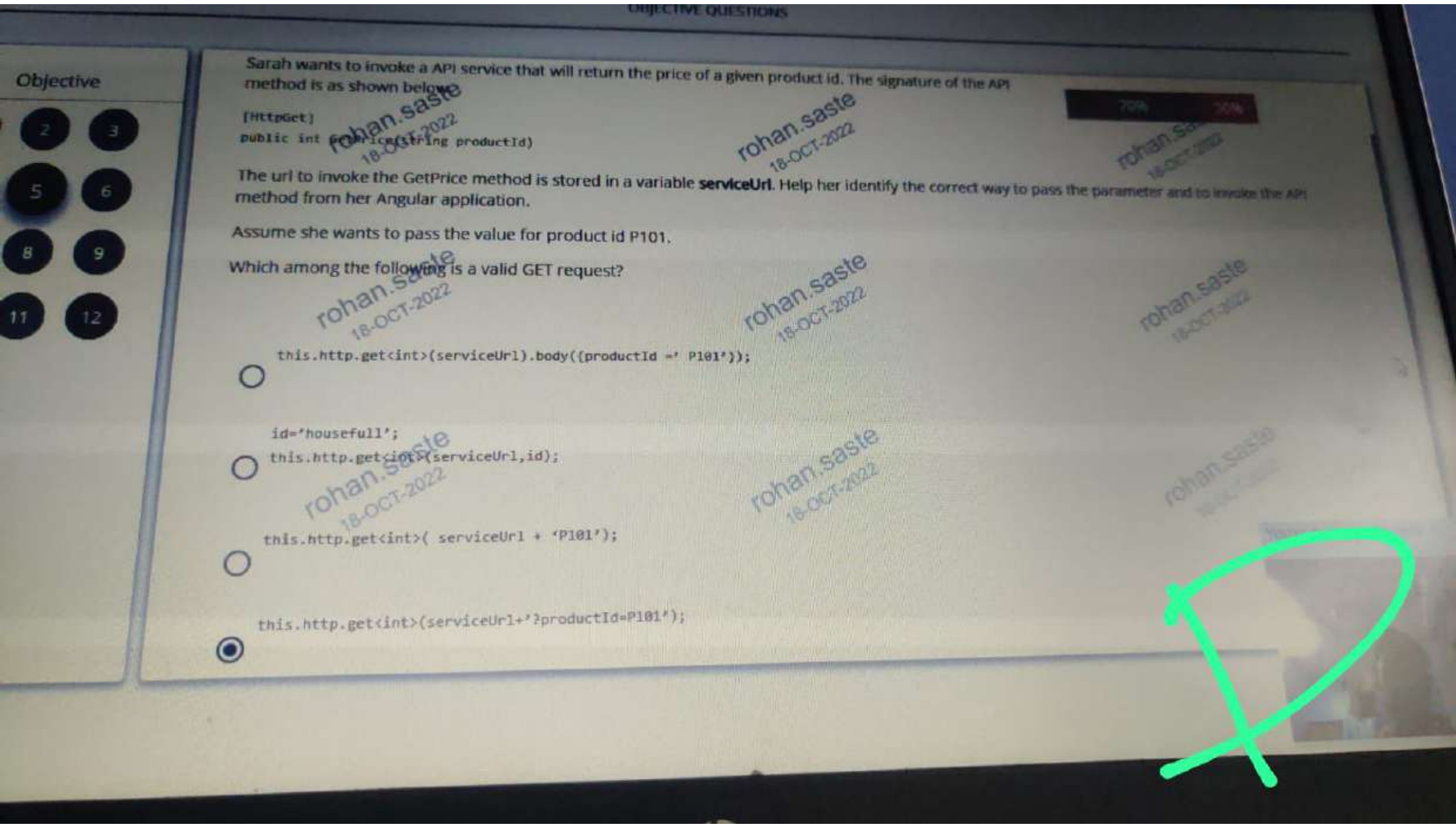
○ `{{ 2021 | currency : 'USD' : true }}`

○ `{{ 2021 | currency : 'USD' : 'code' }}`

○ `{{ 2021 | currency : 'USD' : 'String' }}`

○ `{{ 2021 | currency : 'USD' : 'Dollar' }}`

d



Sarah wants to invoke a API service that will return the price of a given product id. The signature of the API method is as shown below.

```
[HttpGet]
public int GetPrice(string productId)
```

The url to invoke the GetPrice method is stored in a variable **serviceUrl**. Help her identify the correct way to pass the parameter and to invoke the API method from her Angular application.

Assume she wants to pass the value for product id P101.

Which among the following is a valid GET request?

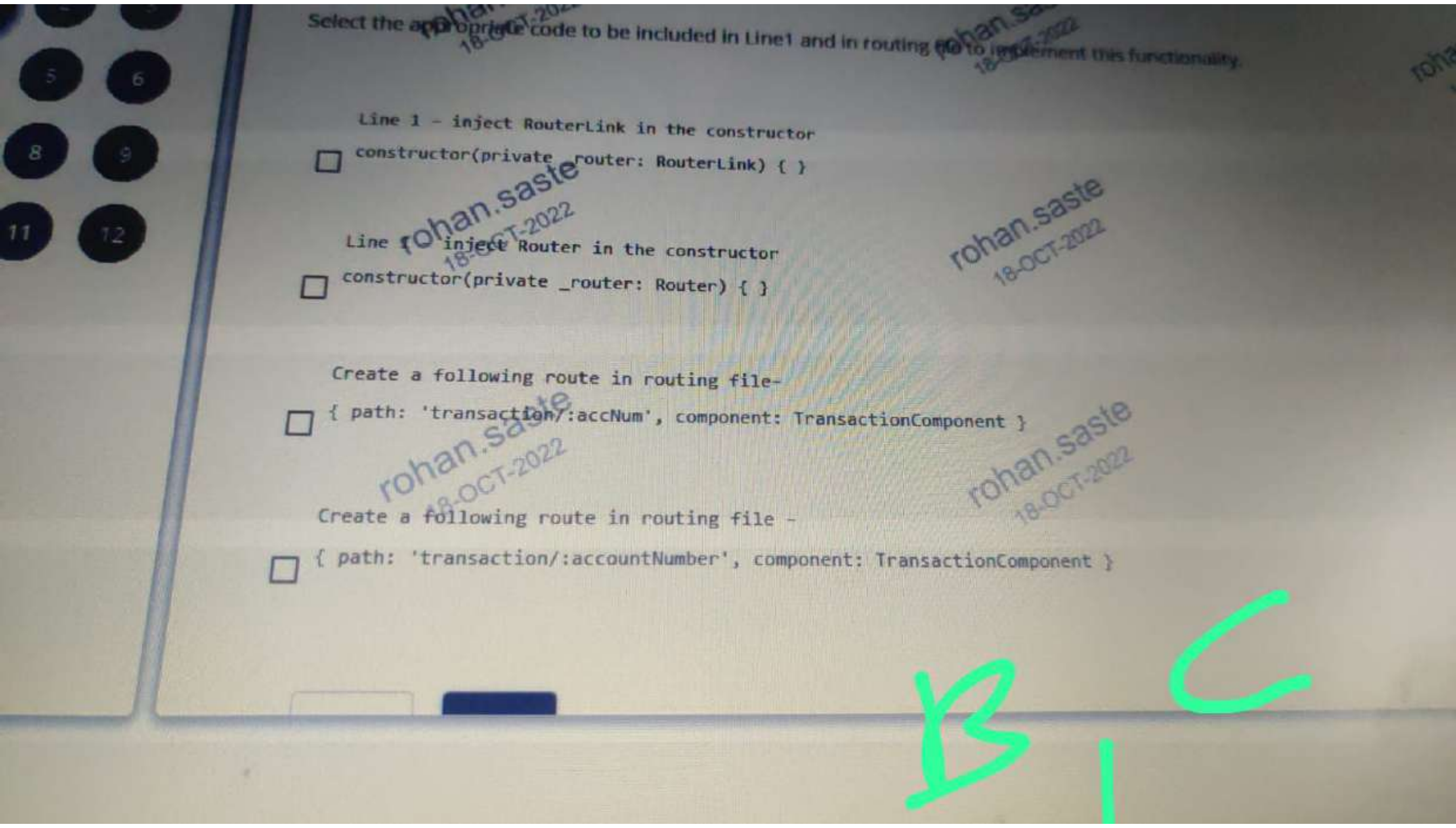
○
```
this.http.get<int>(serviceUrl).body({productId =' P101'});
```

○
```
id='housefull';
this.http.get<int>(serviceUrl,id);
```

○
```
this.http.get<int>( serviceUrl + 'P101');
```

●
```
this.http.get<int>(serviceUrl+'?productId=P101');
```

D



Select the appropriate code to be included in Line1 and in routing file to implement this functionality.

Line 1 - inject RouterLink in the constructor

□ `constructor(private _router: RouterLink) { }`

Line 1 - inject Router in the constructor

□ `constructor(private _router: Router) { }`

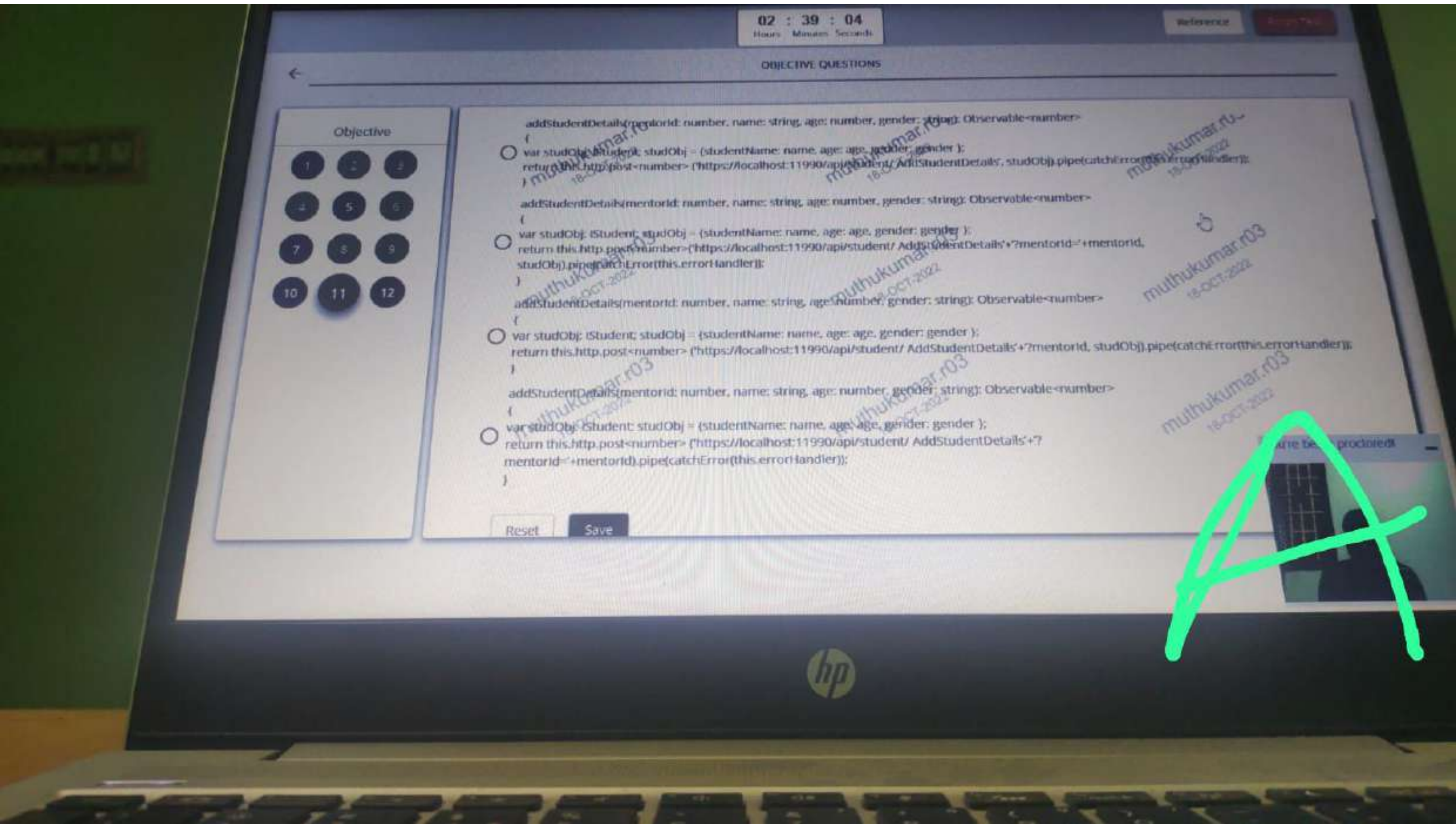
Create a following route in routing file-

□ `{ path: 'transaction/:accNum', component: TransactionComponent }`

Create a following route in routing file -

□ `{ path: 'transaction/:accountNumber', component: TransactionComponent }`

B, C



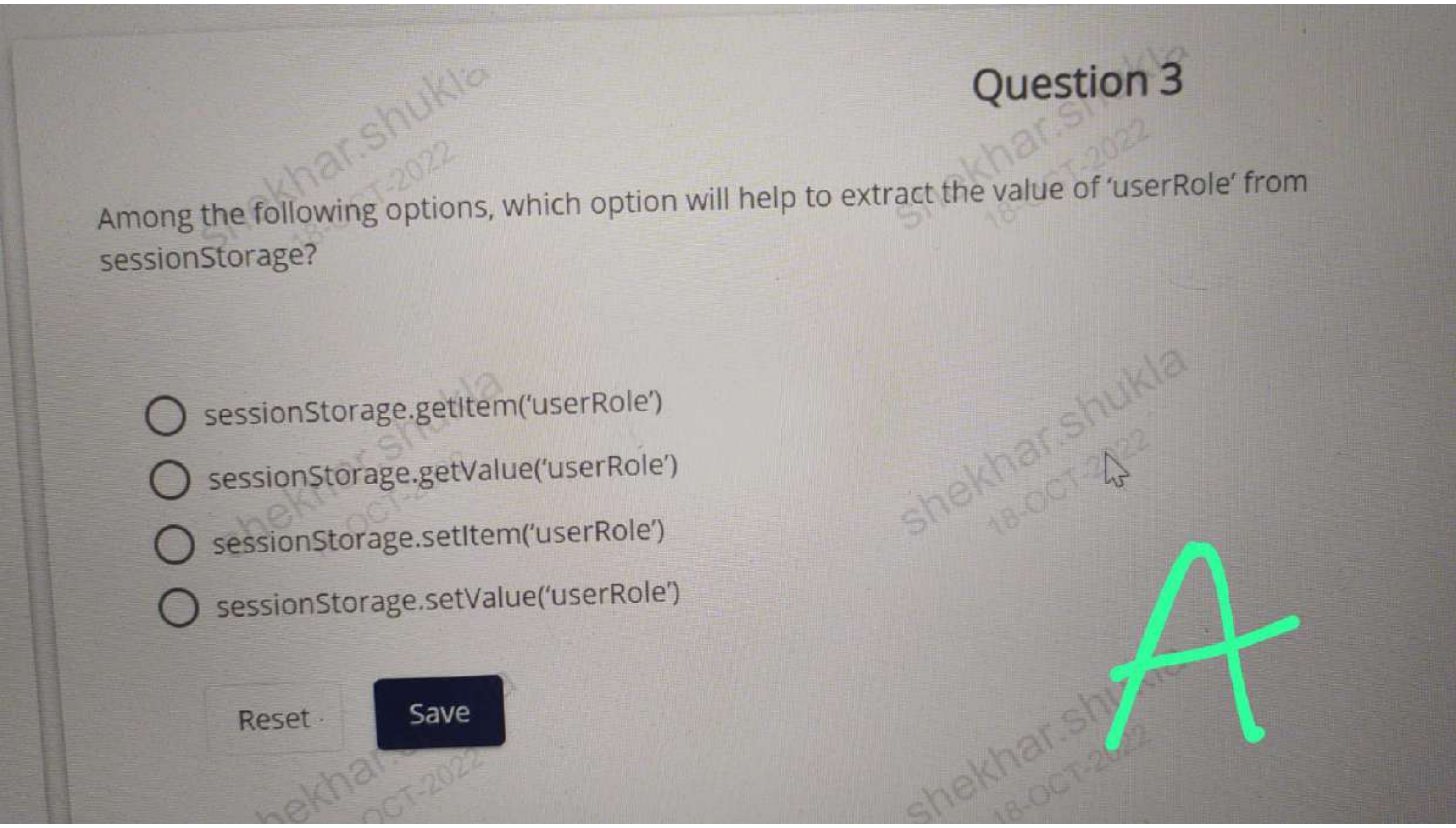
○
```
addStudentDetails(mentorId: number, name: string, age: number, gender: string): Observable<number>
{
var studObj: IStudent; studObj = {studentName: name, age: age, gender: gender };
return this.http.post<number> ('https://localhost:11990/api/student/ AddStudentDetails', studObj).pipe(catchError(this.errorHandler));
}
```

○
```
addStudentDetails(mentorId: number, name: string, age: number, gender: string): Observable<number>
{
var studObj: IStudent; studObj = {studentName: name, age: age, gender: gender };
return this.http.post<number>('https://localhost:11990/api/student/ AddStudentDetails'+'?mentorId='+mentorId,
studObj).pipe(catchError(this.errorHandler));
}
```

○
```
addStudentDetails(mentorId: number, name: string, age: number, gender: string): Observable<number>
{
var studObj: IStudent; studObj = {studentName: name, age: age, gender: gender };
return this.http.post<number> ('https://localhost:11990/api/student/ AddStudentDetails'+'?mentorId, studObj).pipe(catchError(this.errorHandler));
}
```

○
```
addStudentDetails(mentorId: number, name: string, age: number, gender: string): Observable<number>
{
var studObj: IStudent; studObj = {studentName: name, age: age, gender: gender };
return this.http.post<number> ('https://localhost:11990/api/student/ AddStudentDetails'+'?
mentorId='+mentorId).pipe(catchError(this.errorHandler));
}
```

A



Question 3

Among the following options, which option will help to extract the value of 'userRole' from sessionStorage?

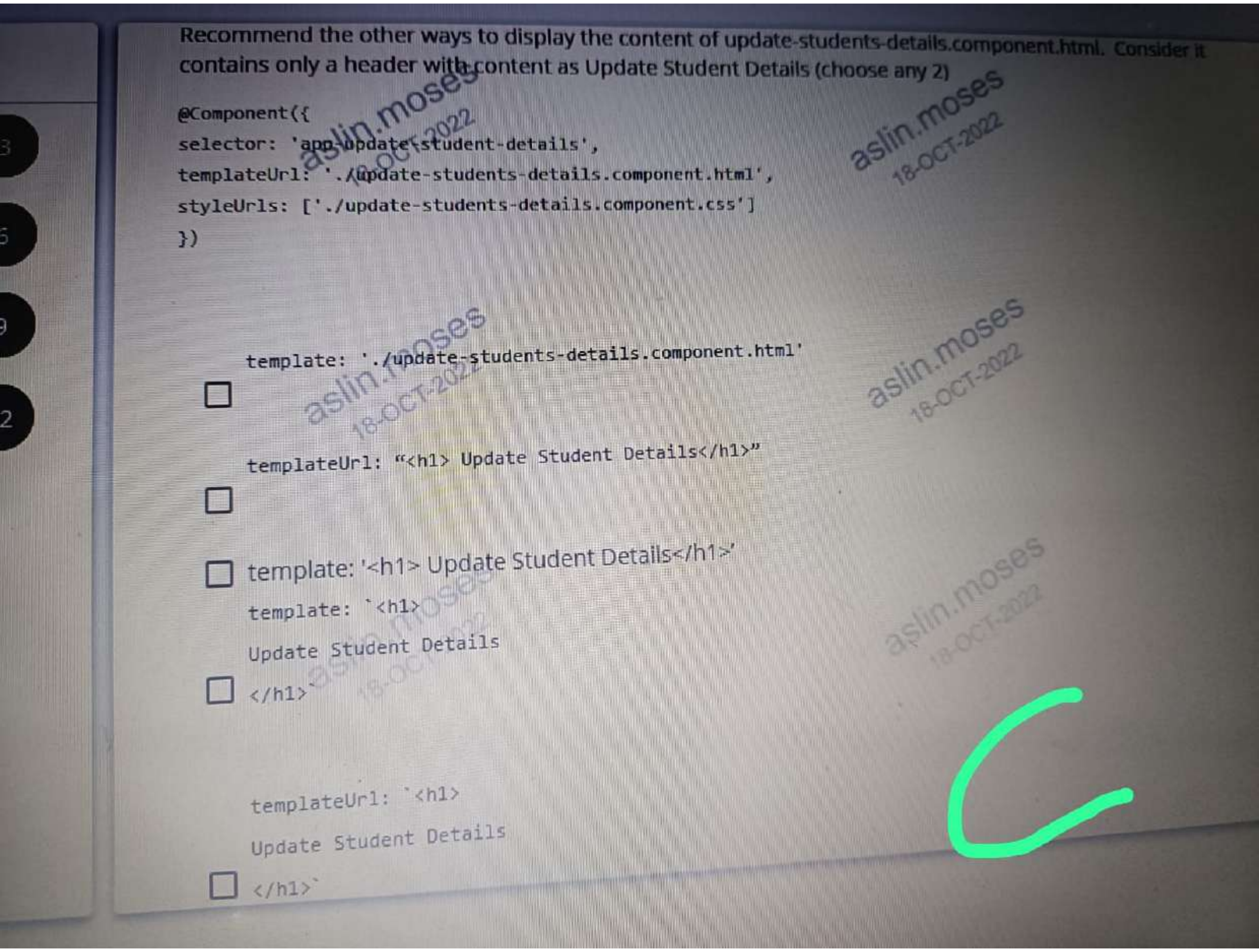
○ sessionStorage.getItem('userRole')

○ sessionStorage.getValue('userRole')

○ sessionStorage.setItem('userRole')

○ sessionStorage.setValue('userRole')

A



Recommend the other ways to display the content of update-students-details.component.html. Consider it contains only a header with content as Update Student Details (choose any 2)

```
@Component({
selector: 'app-update-student-details',
templateUrl: './update-students-details.component.html',
styleUrls: ['./update-students-details.component.css']
})
```

□
```
template: './update-students-details.component.html'
```

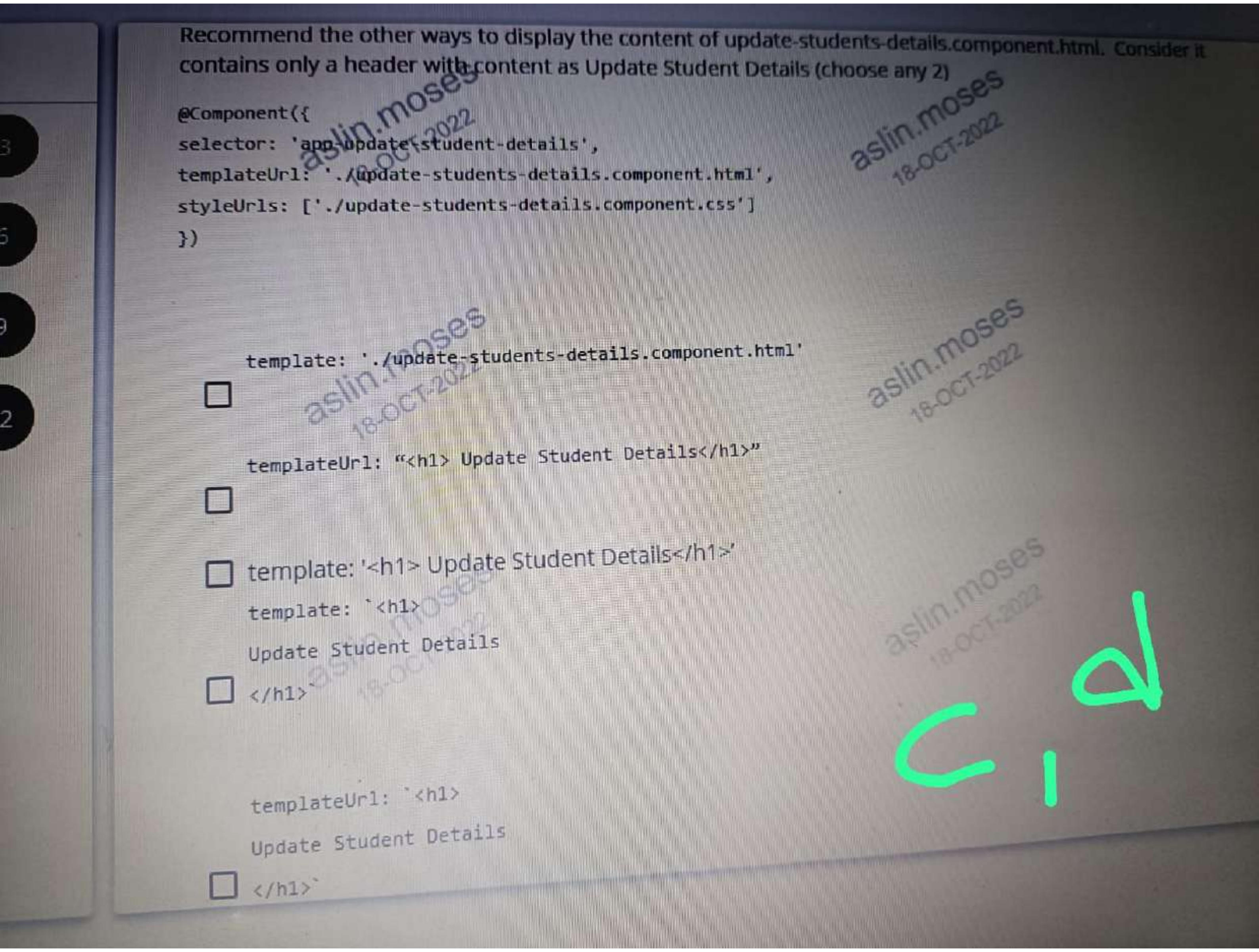
□
```
templateUrl: “<h1> Update Student Details</h1>”
```

□ template: '<h1> Update Student Details</h1>'

□
```
template: `<h1>
Update Student Details
</h1>`
```

□
```
templateUrl: `<h1>
Update Student Details
</h1>`
```

C



Recommend the other ways to display the content of update-students-details.component.html. Consider it contains only a header with content as Update Student Details (choose any 2)

```
@Component({
selector: 'app-update-student-details',
templateUrl: './update-students-details.component.html',
styleUrls: ['./update-students-details.component.css']
})
```

□ template: './update-students-details.component.html'

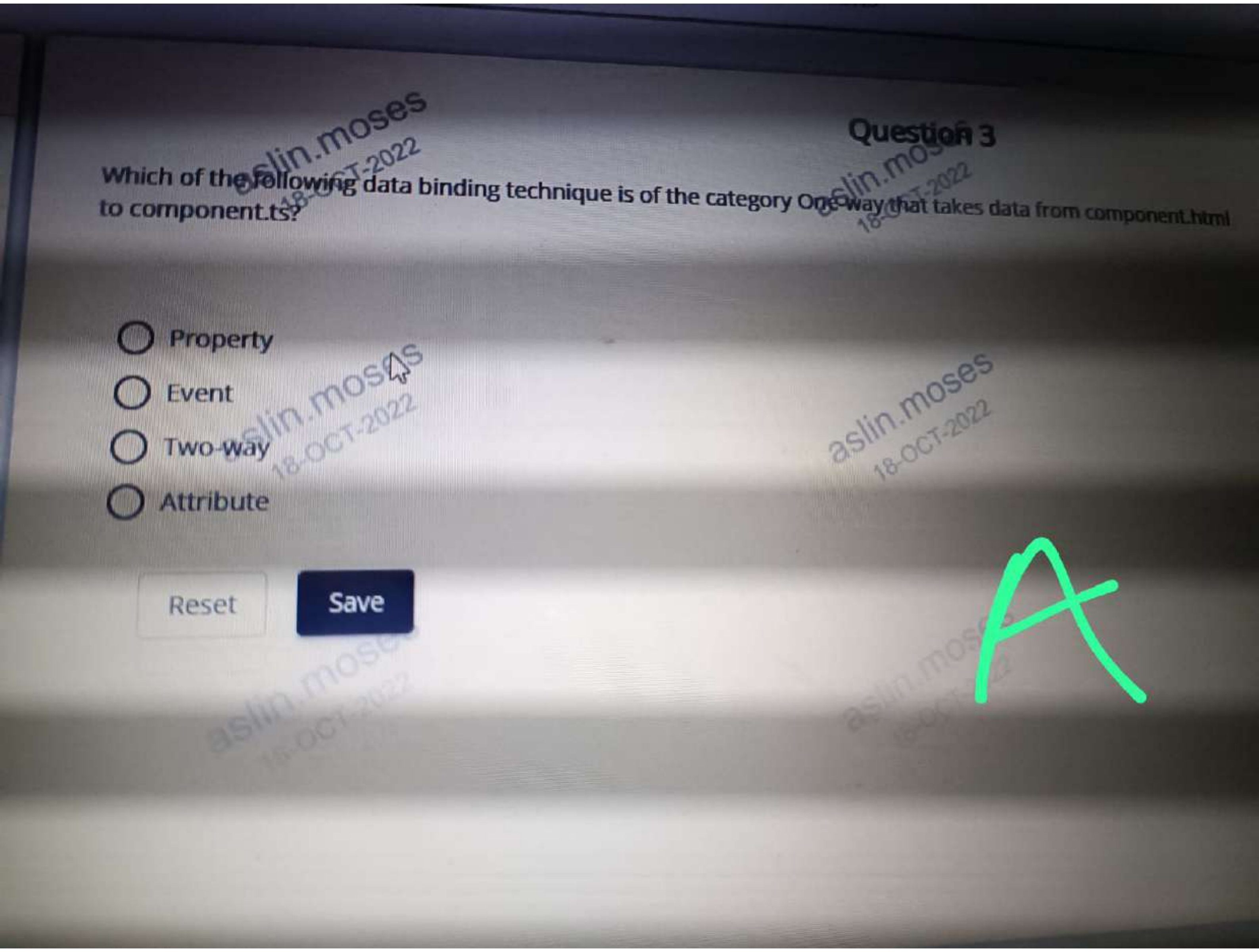
□ templateUrl: "<h1> Update Student Details</h1>"

□ template: '<h1> Update Student Details</h1>'

□ template: `<h1>
Update Student Details
</h1>`

□ templateUrl: `<h1>
Update Student Details
</h1>`

c, d



Question 3

Which of the following data binding technique is of the category One-way that takes data from component.html to component.ts?

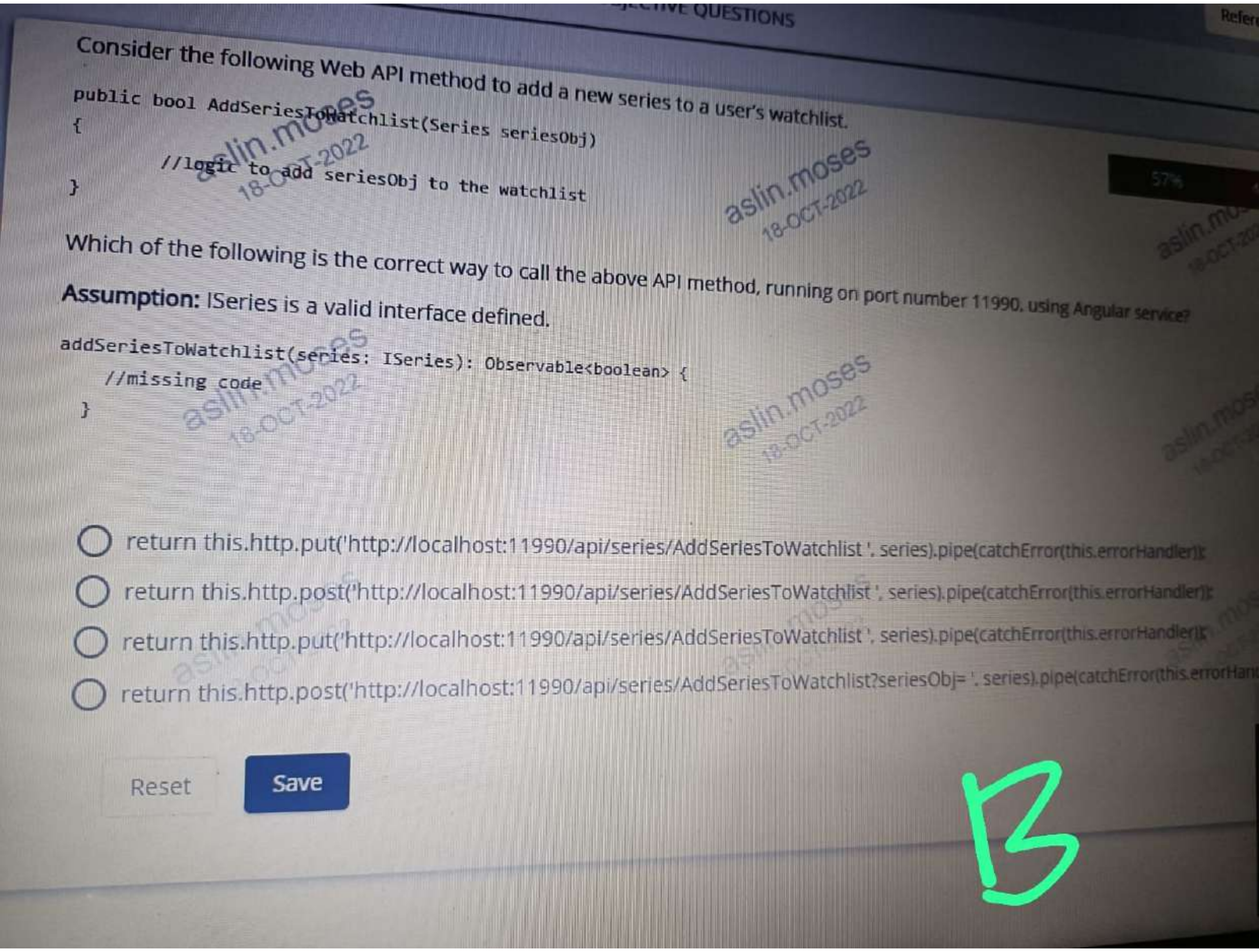
○ Property

○ Event

○ Two-way

○ Attribute

A



Consider the following Web API method to add a new series to a user's watchlist.

```
public bool AddSeriesToWatchlist(Series seriesObj)
{
        //logic to add seriesObj to the watchlist
}
```

Which of the following is the correct way to call the above API method, running on port number 11990, using Angular service?

Assumption: ISeries is a valid interface defined.

```
addSeriesToWatchlist(series: ISeries): Observable<boolean> {
    //missing code
 }
```

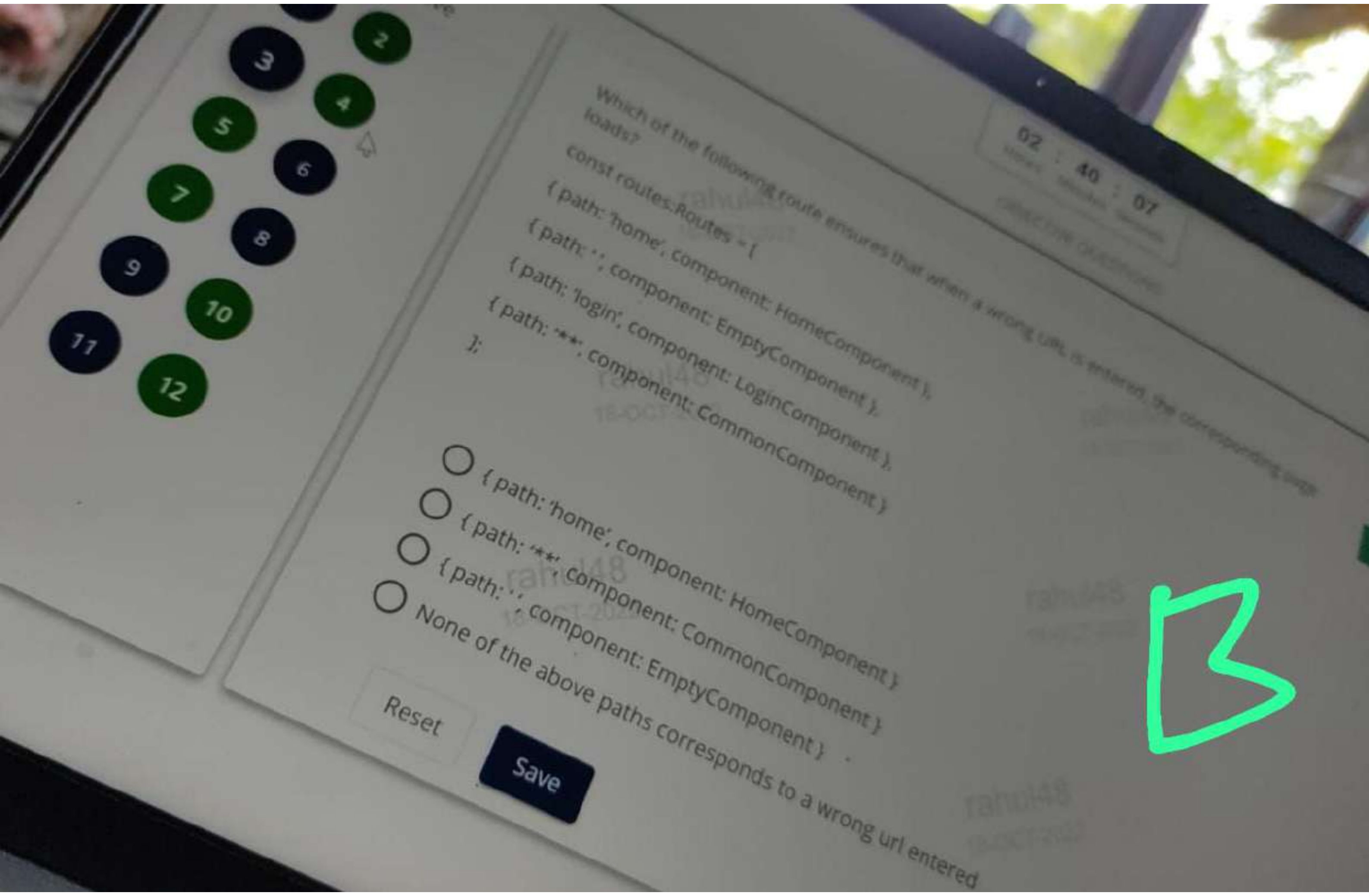
○ return this.http.put('http://localhost:11990/api/series/AddSeriesToWatchlist ', series).pipe(catchError(this.errorHandler));

○ return this.http.post('http://localhost:11990/api/series/AddSeriesToWatchlist ', series).pipe(catchError(this.errorHandler));

○ return this.http.put('http://localhost:11990/api/series/AddSeriesToWatchlist ', series).pipe(catchError(this.errorHandler));

○ return this.http.post('http://localhost:11990/api/series/AddSeriesToWatchlist?seriesObj= ', series).pipe(catchError(this.errorHan

B



Which of the following route ensures that when a wrong URL is entered, the corresponding page loads?

```
const routes:Routes = [
{ path: 'home', component: HomeComponent },
{ path: '', component: EmptyComponent },
{ path: 'login', component: LoginComponent },
{ path: '**', component: CommonComponent }
];
```

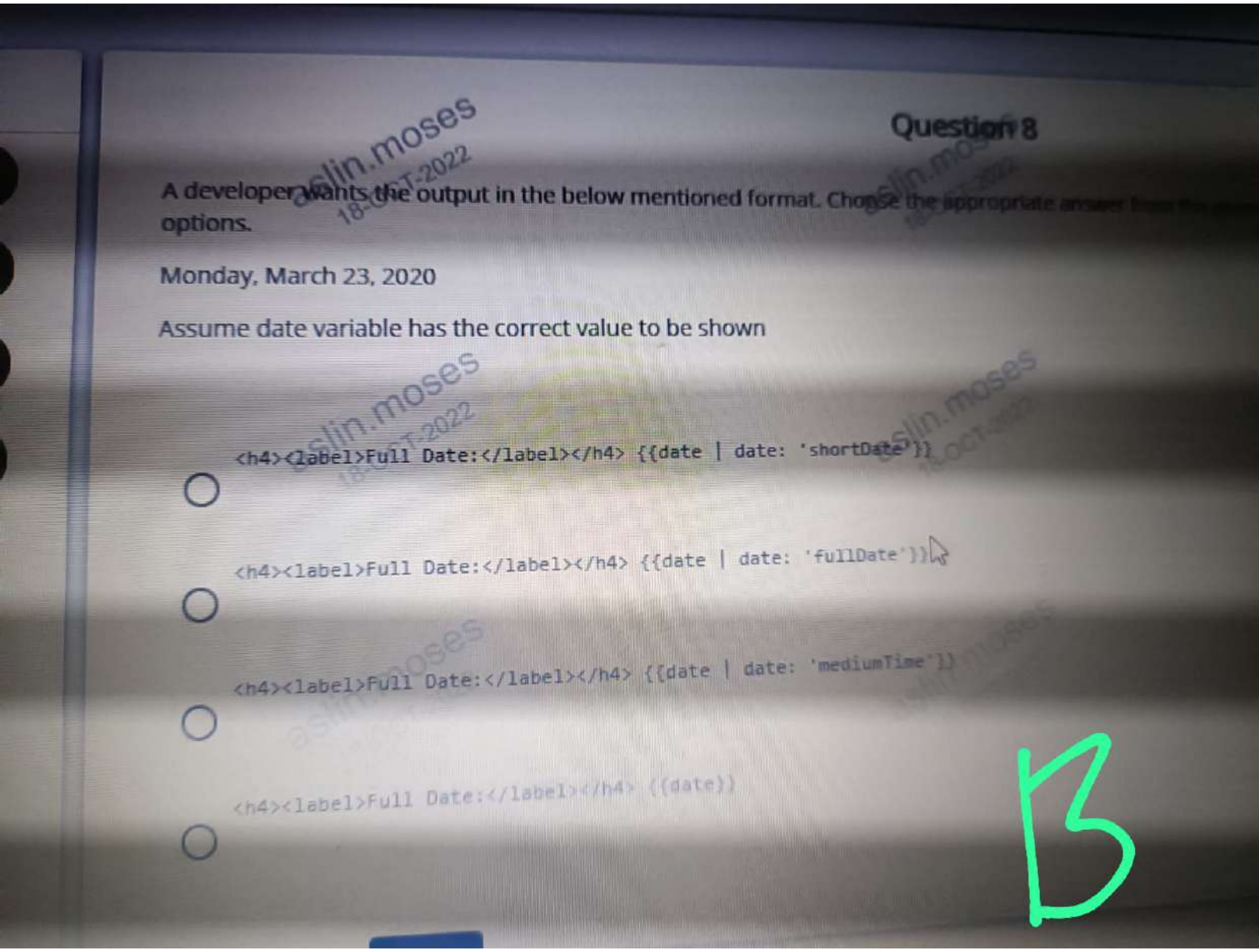
○ { path: 'home', component: HomeComponent }

○ { path: '**', component: CommonComponent }

○ { path: '', component: EmptyComponent }

○ None of the above paths corresponds to a wrong url entered

B



Question 8

A developer wants the output in the below mentioned format. Choose the appropriate answer from the given options.

Monday, March 23, 2020

Assume date variable has the correct value to be shown

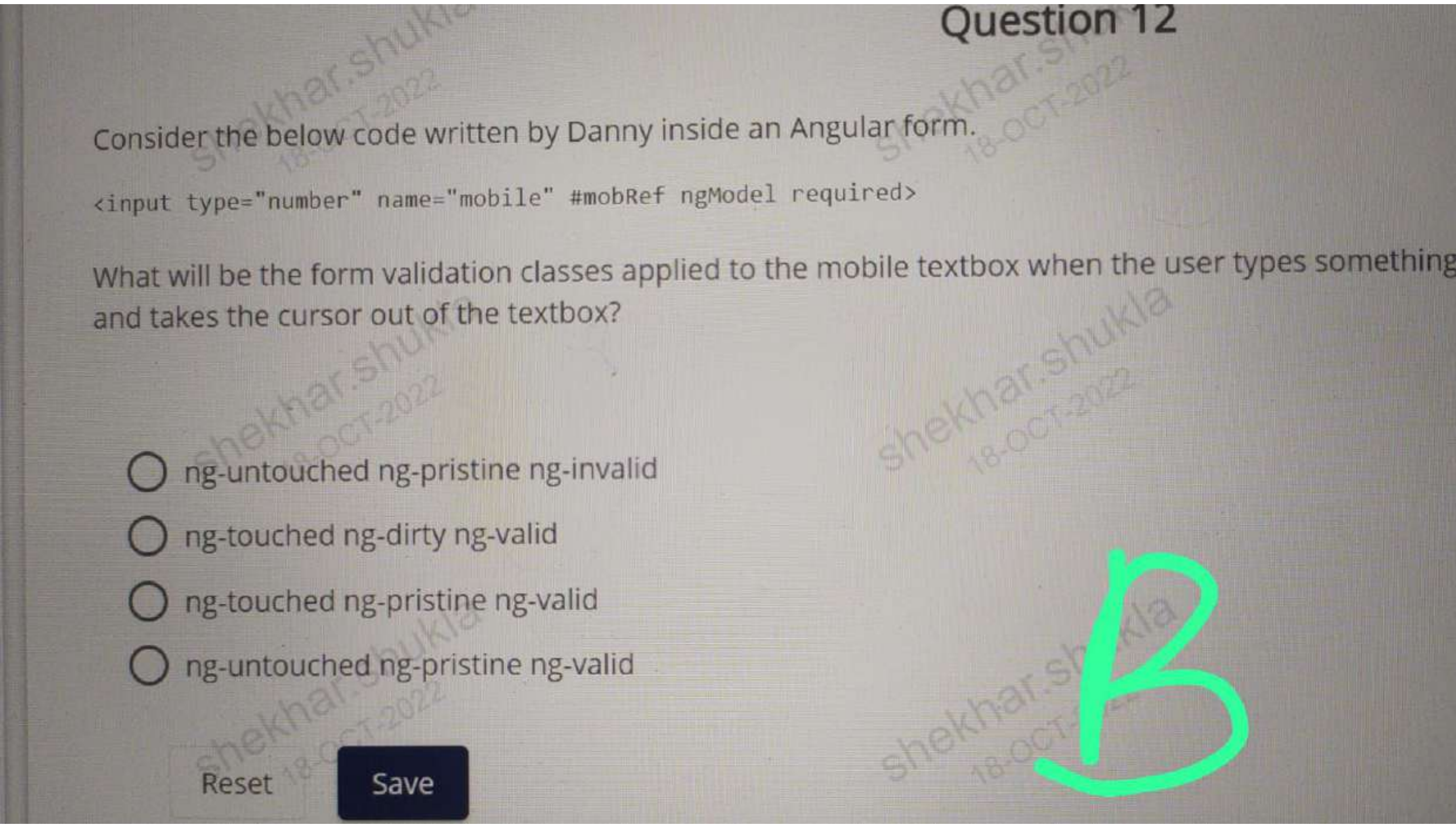
○ `<h4><label>Full Date:</label></h4> {{date | date: 'shortDate'}}`

○ `<h4><label>Full Date:</label></h4> {{date | date: 'fullDate'}}`

○ `<h4><label>Full Date:</label></h4> {{date | date: 'mediumTime'}}`

○ `<h4><label>Full Date:</label></h4> {{date}}`

B



Question 12

Consider the below code written by Danny inside an Angular form.

```
<input type="number" name="mobile" #mobRef ngModel required>
```

What will be the form validation classes applied to the mobile textbox when the user types something and takes the cursor out of the textbox?

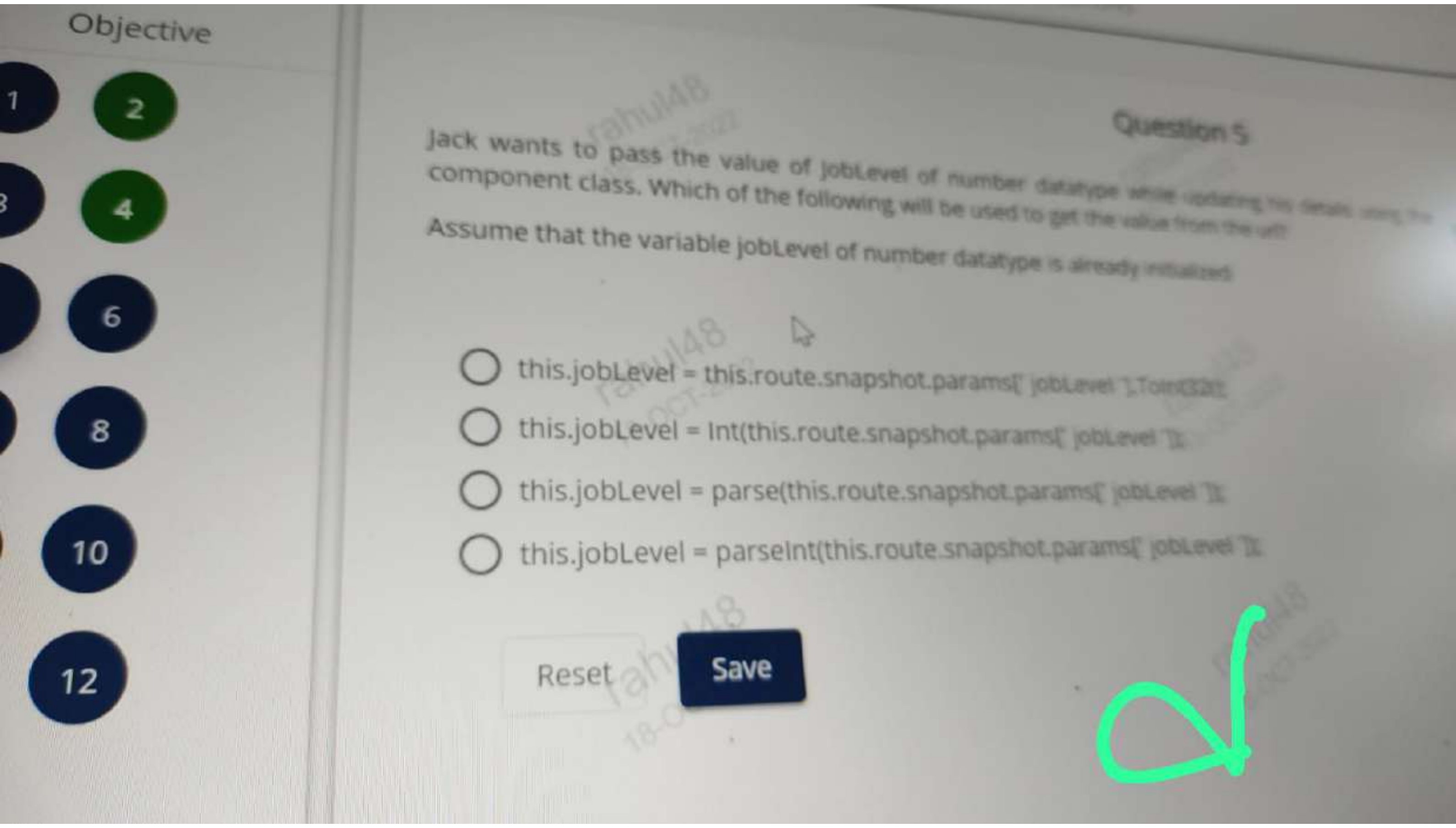
○ ng-untouched ng-pristine ng-invalid

○ ng-touched ng-dirty ng-valid

○ ng-touched ng-pristine ng-valid

○ ng-untouched ng-pristine ng-valid

B



Question 5

Jack wants to pass the value of jobLevel of number datatype while updating his details using the component class. Which of the following will be used to get the value from the url?

Assume that the variable jobLevel of number datatype is already initialized

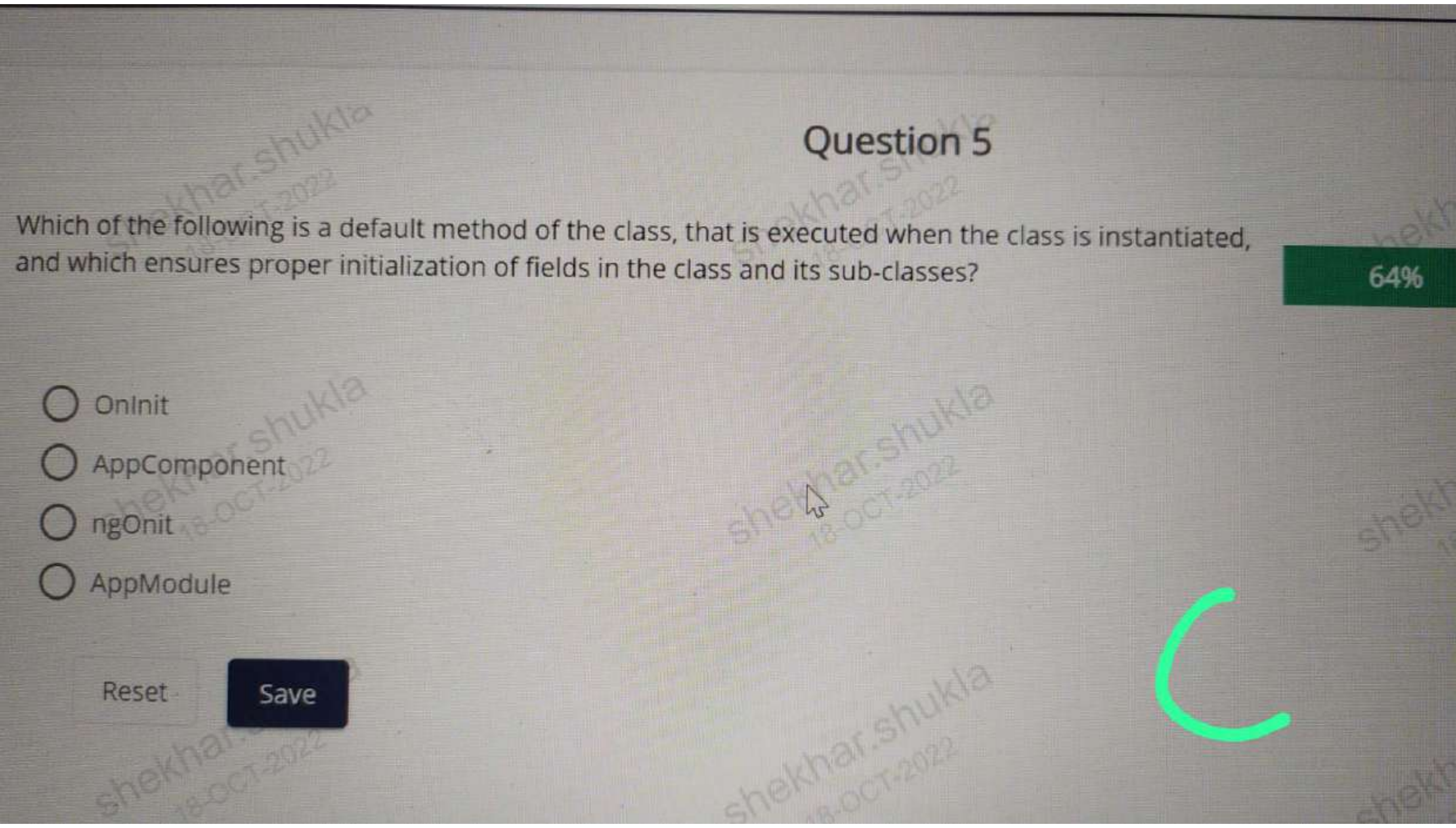
○ this.jobLevel = this.route.snapshot.params[' jobLevel '].ToInt32();

○ this.jobLevel = Int(this.route.snapshot.params[' jobLevel ']);

○ this.jobLevel = parse(this.route.snapshot.params[' jobLevel ']);

○ this.jobLevel = parseInt(this.route.snapshot.params[' jobLevel ']);

d



Question 5

Which of the following is a default method of the class, that is executed when the class is instantiated, and which ensures proper initialization of fields in the class and its sub-classes?

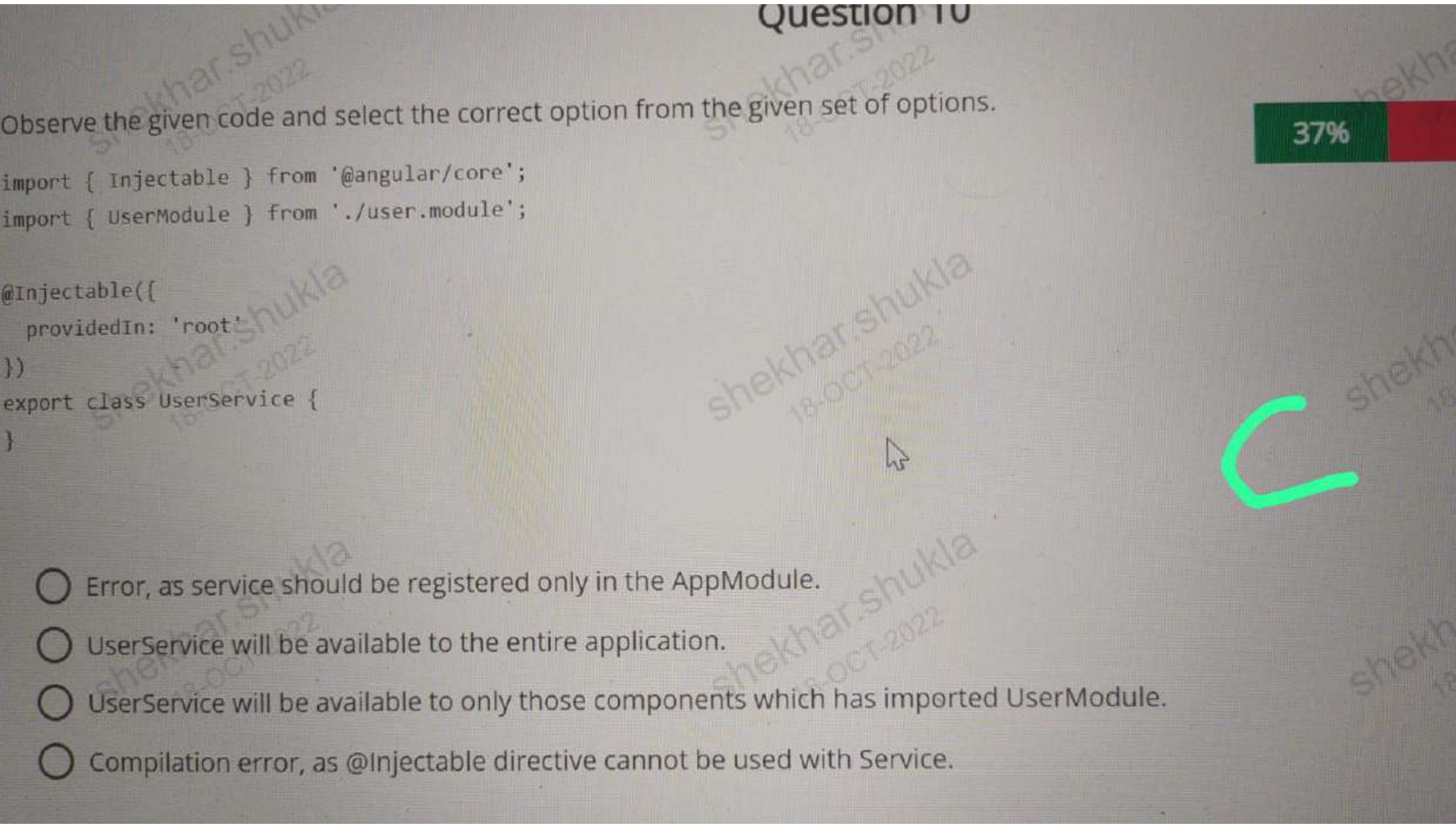
○ OnInit

○ AppComponent

○ ngOnit

○ AppModule

C



Question 10

37%

Observe the given code and select the correct option from the given set of options.

```
import { Injectable } from '@angular/core';
import { UserModule } from './user.module';

@Injectable({
  providedIn: 'root'
})
export class UserService {
}
```

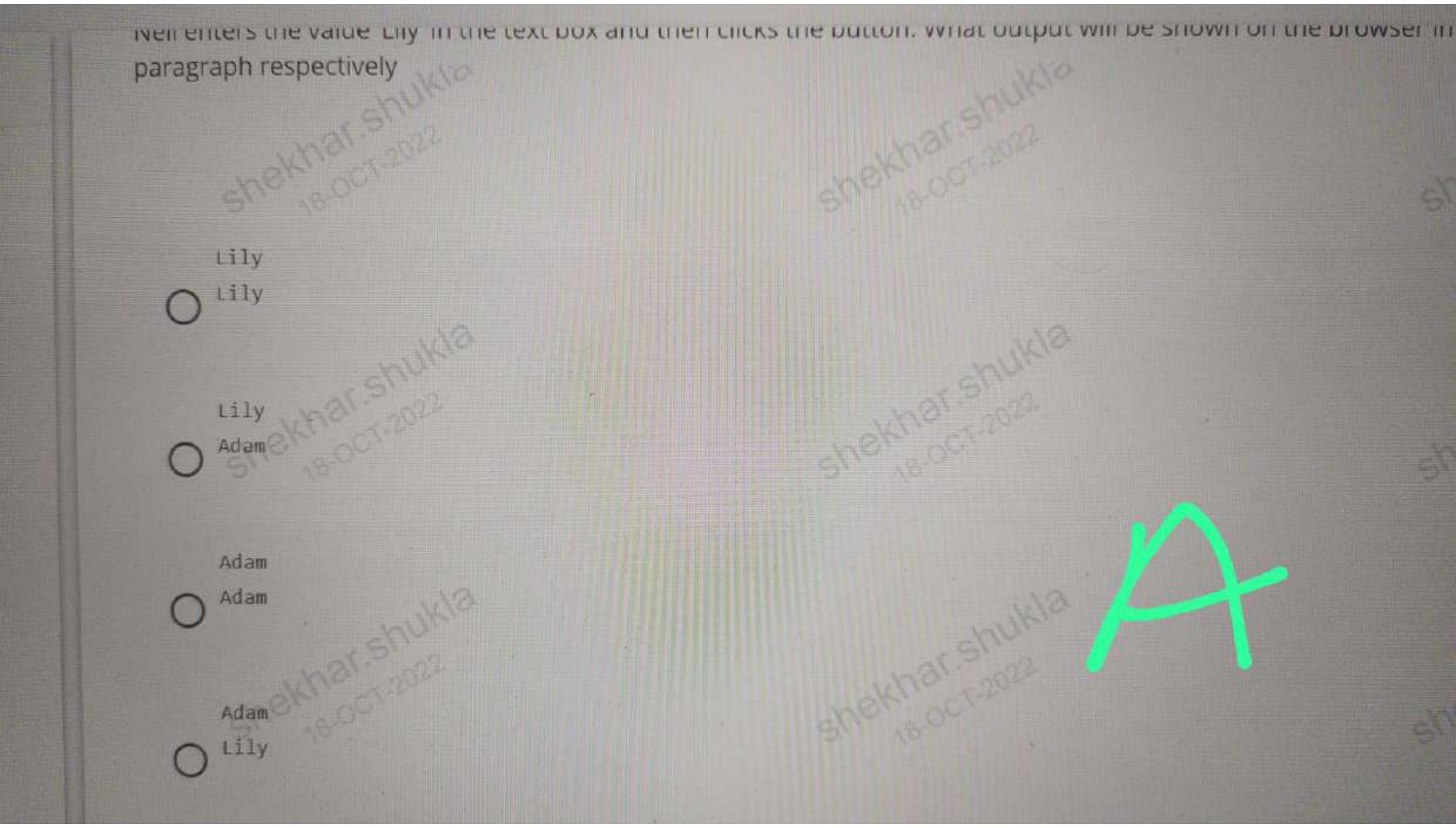
C

○ Error, as service should be registered only in the AppModule.

○ UserService will be available to the entire application.

○ UserService will be available to only those components which has imported UserModule.

○ Compilation error, as @Injectable directive cannot be used with Service.



Neil enters the value 'Lily' in the text box and then clicks the button. What output will be shown on the browser in
paragraph respectively

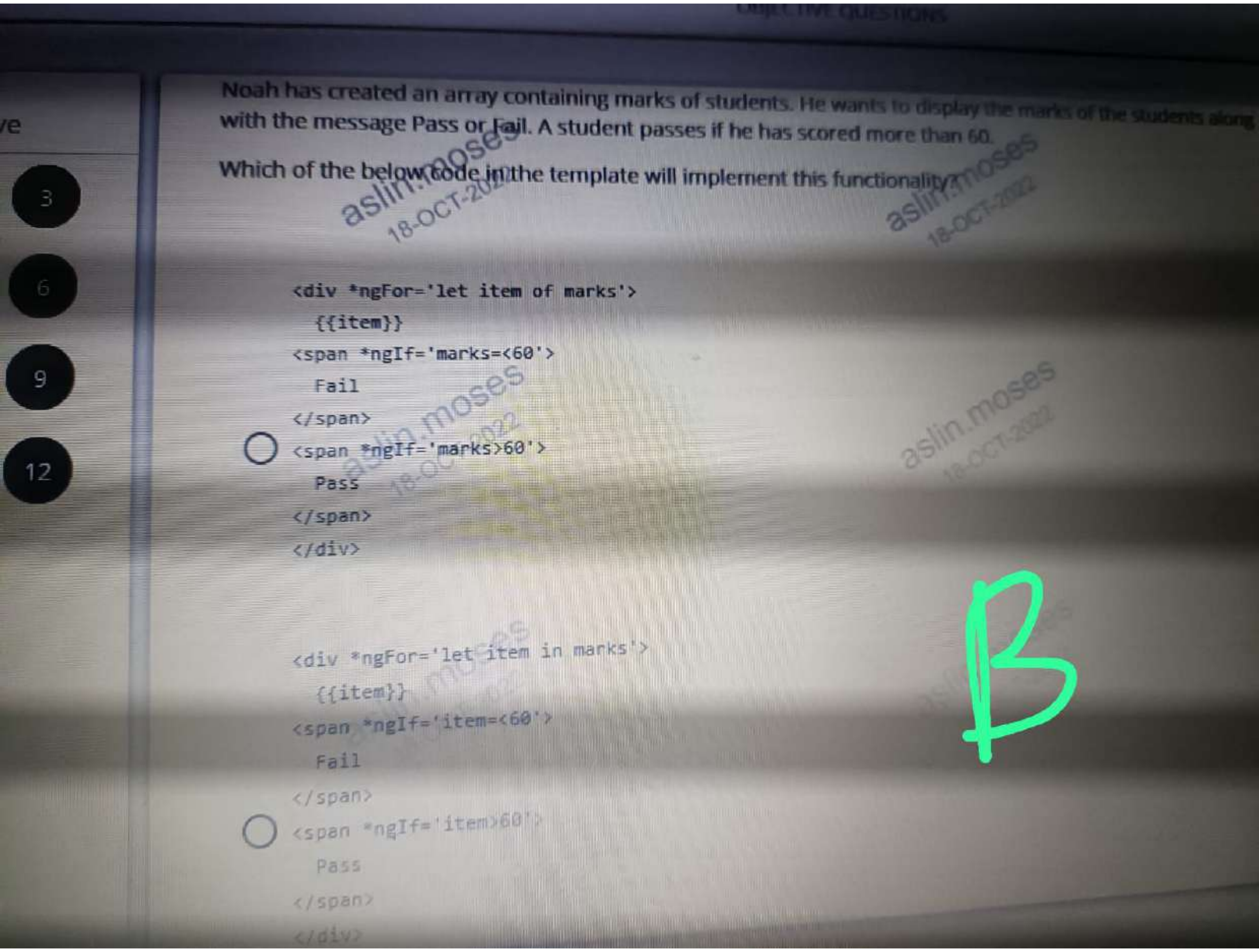
○ Lily
Lily

○ Lily
Adam

○ Adam
Adam

○ Adam
Lily

A



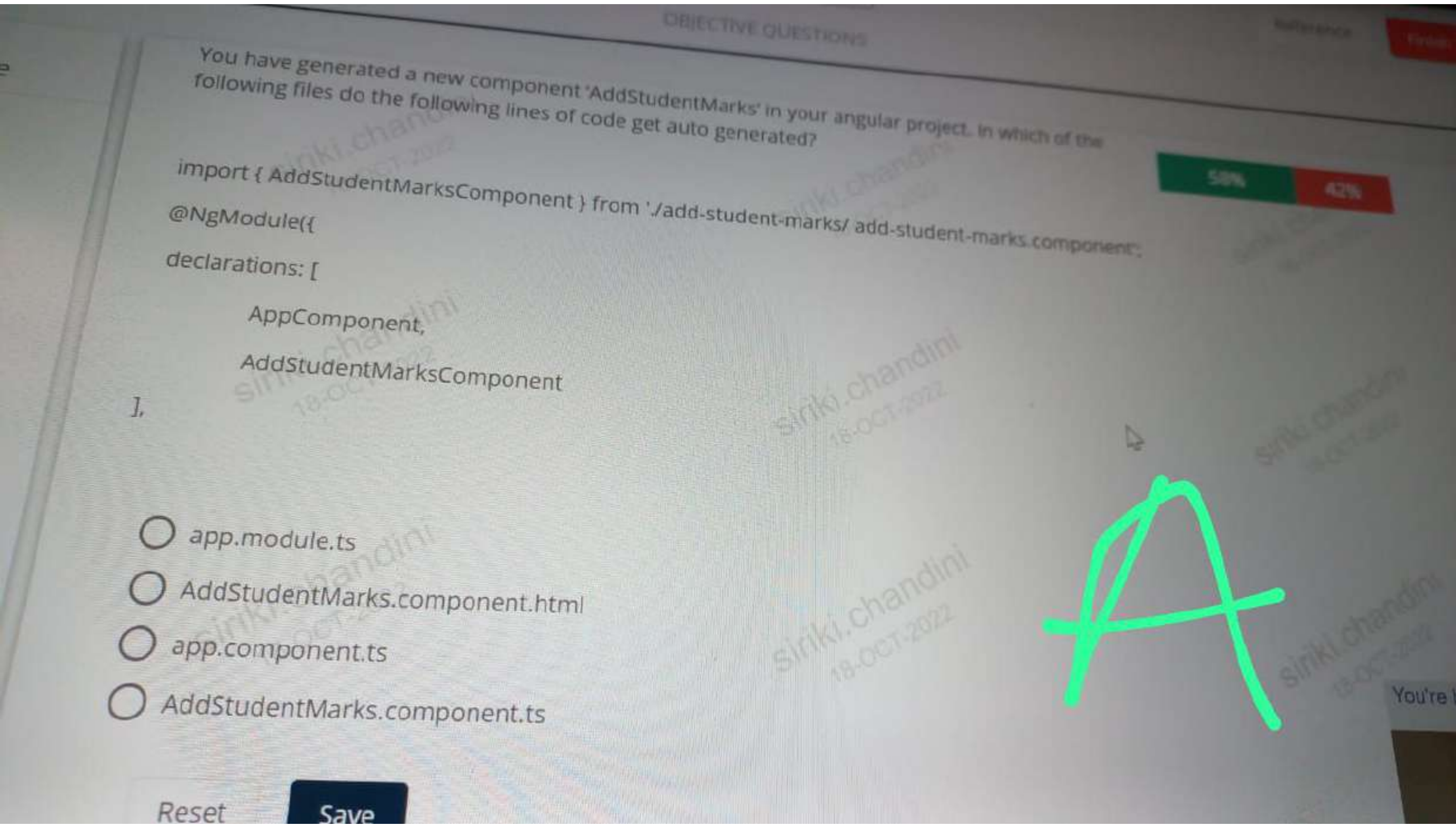
Noah has created an array containing marks of students. He wants to display the marks of the students along with the message Pass or Fail. A student passes if he has scored more than 60.

Which of the below code in the template will implement this functionality?

○
```
<div *ngFor='let item of marks'>
  {{item}}
<span *ngIf='marks=<60'>
  Fail
</span>
<span *ngIf='marks>60'>
  Pass
</span>
</div>
```

○
```
<div *ngFor='let item in marks'>
  {{item}}
<span *ngIf='item=<60'>
  Fail
</span>
<span *ngIf='item>60'>
  Pass
</span>
</div>
```

B



You have generated a new component 'AddStudentMarks' in your angular project. In which of the following files do the following lines of code get auto generated?

```
import { AddStudentMarksComponent } from './add-student-marks/ add-student-marks.component';
@NgModule({
declarations: [
        AppComponent,
        AddStudentMarksComponent
],
```

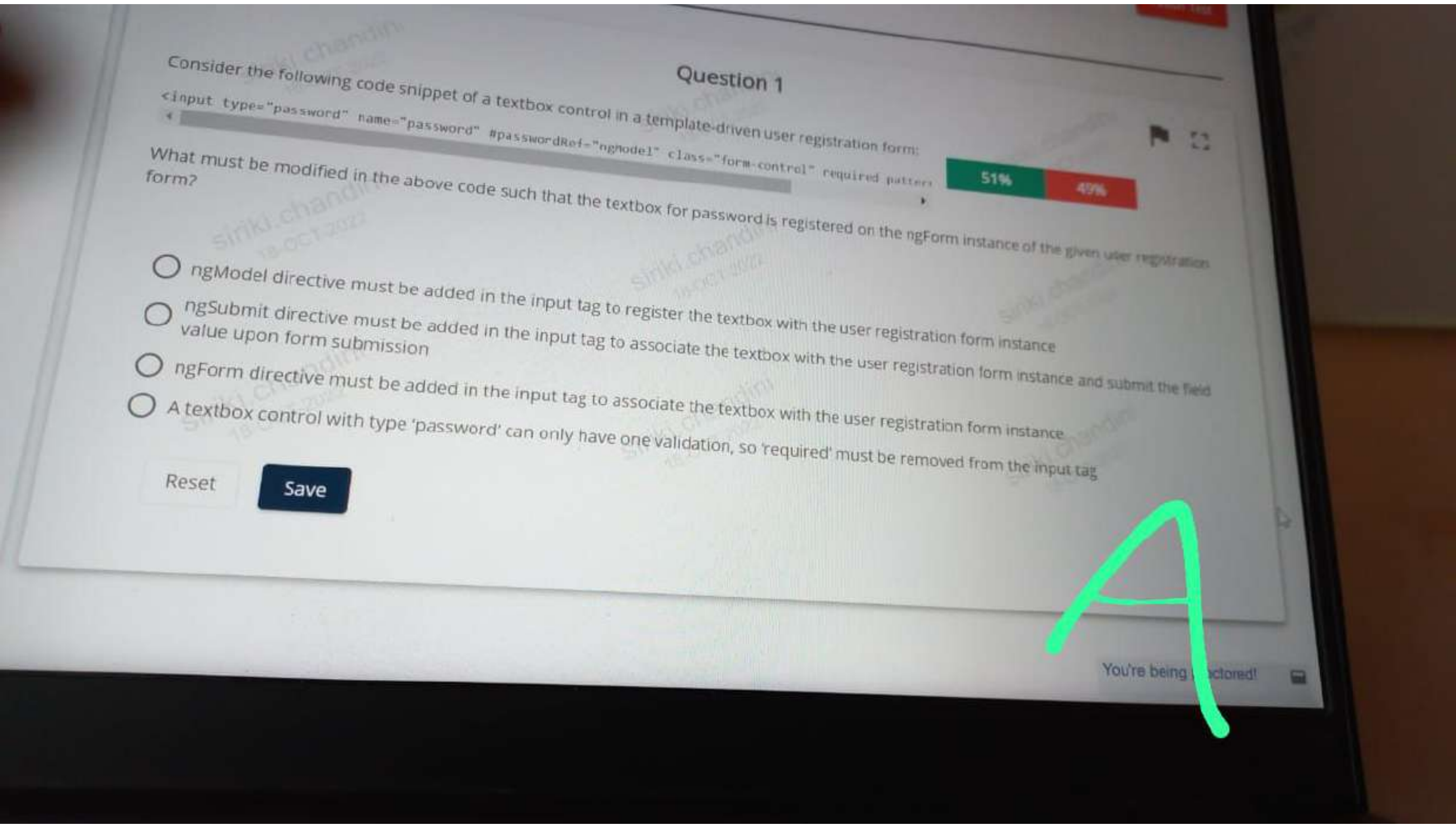
○ app.module.ts

○ AddStudentMarks.component.html

○ app.component.ts

○ AddStudentMarks.component.ts

A



Question 1

Consider the following code snippet of a textbox control in a template-driven user registration form:

```
<input type="password" name="password" #passwordRef="ngModel" class="form-control" required patter
```

What must be modified in the above code such that the textbox for password is registered on the ngForm instance of the given user registration form?

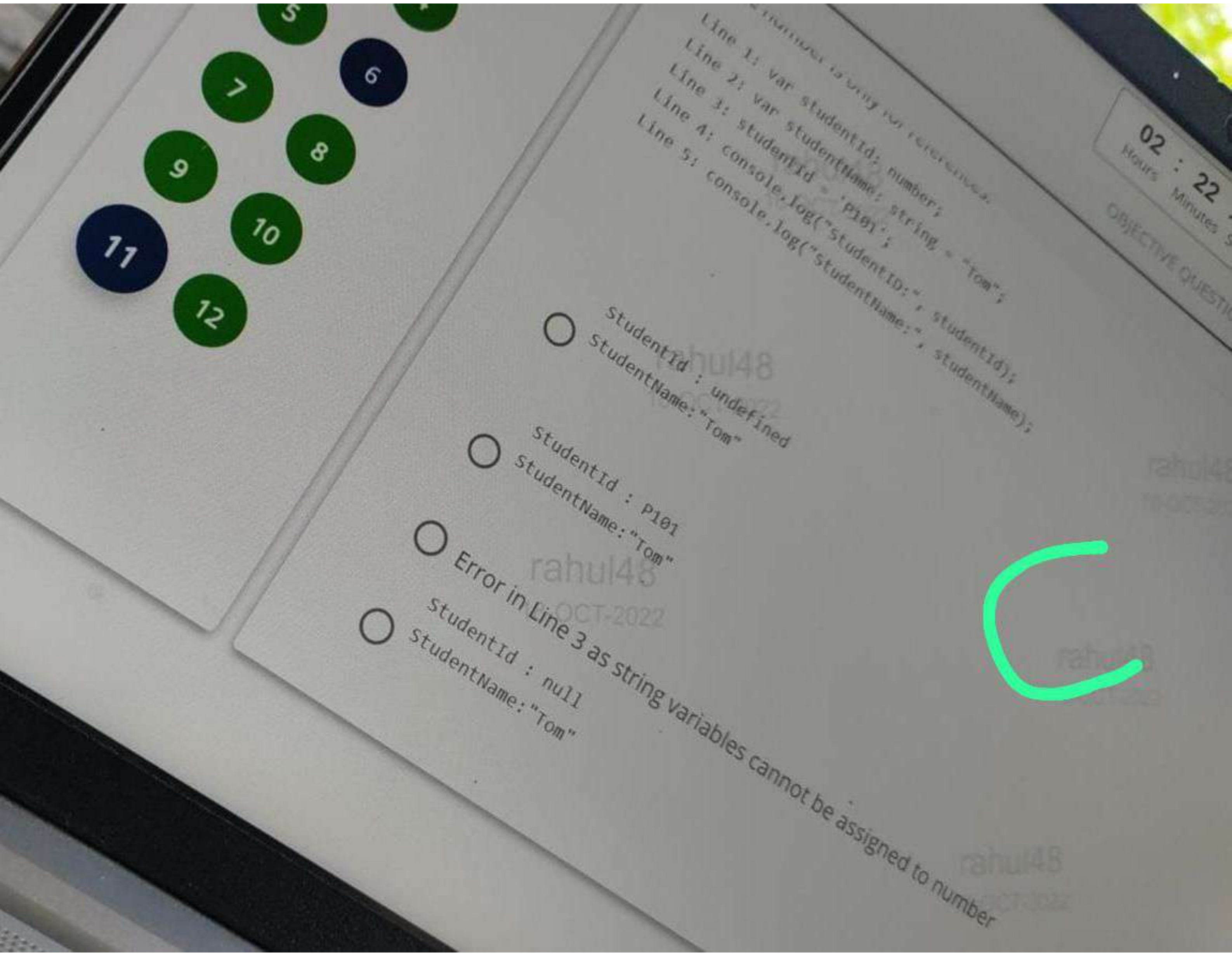
○ ngModel directive must be added in the input tag to register the textbox with the user registration form instance

○ ngSubmit directive must be added in the input tag to associate the textbox with the user registration form instance and submit the field value upon form submission

○ ngForm directive must be added in the input tag to associate the textbox with the user registration form instance

○ A textbox control with type 'password' can only have one validation, so 'required' must be removed from the input tag

A



number is only for reference.

```
Line 1: var studentId: number;
Line 2: var studentName: string = "Tom";
Line 3: studentId = 'P101';
Line 4: console.log("StudentID:", studentId);
Line 5: console.log("StudentName:", studentName);
```

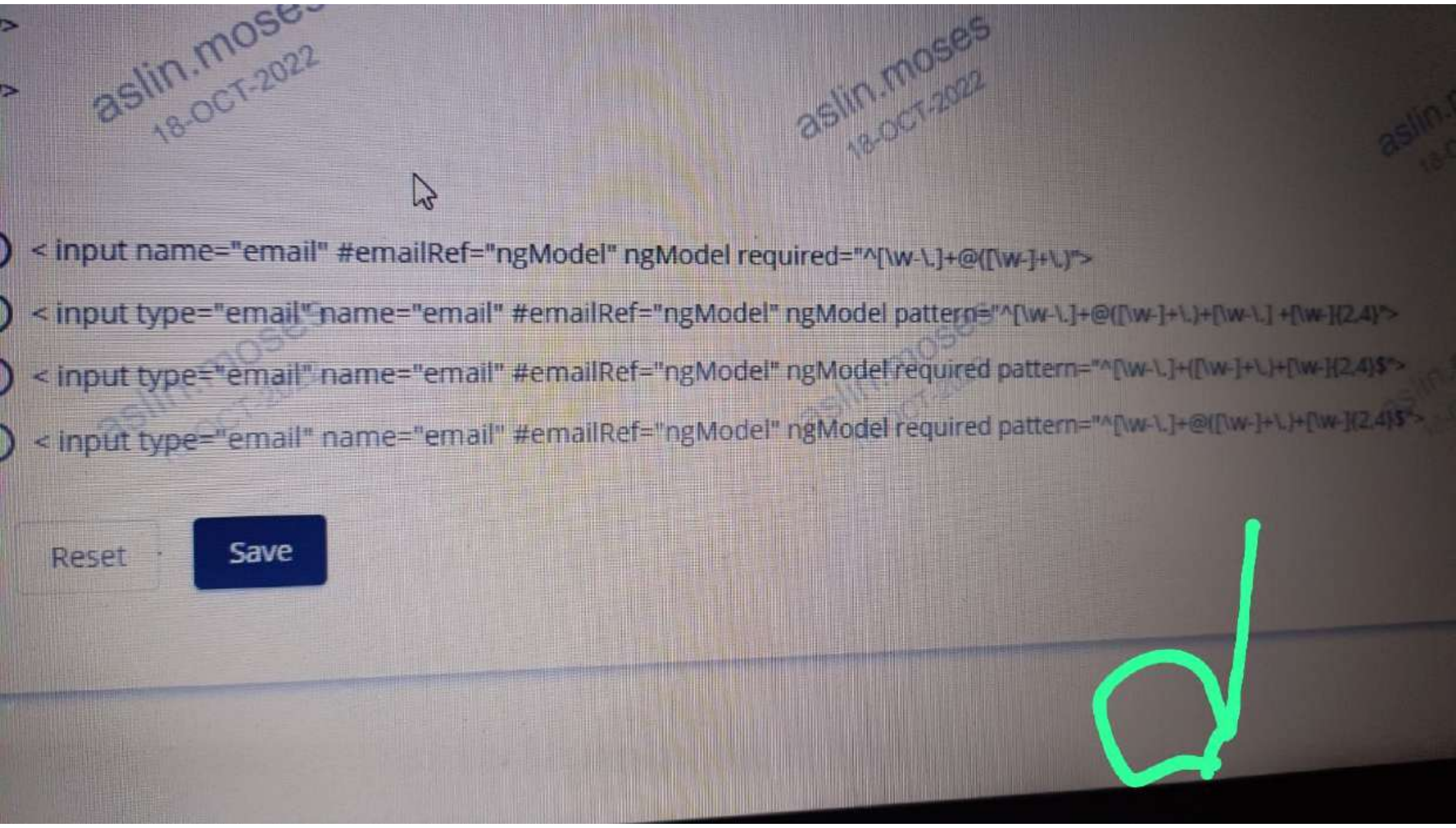
○
```
StudentId : undefined
StudentName:"Tom"
```

○
```
StudentId : P101
StudentName:"Tom"
```

○ Error in Line 3 as string variables cannot be assigned to number

○
```
StudentId : null
StudentName:"Tom"
```

C



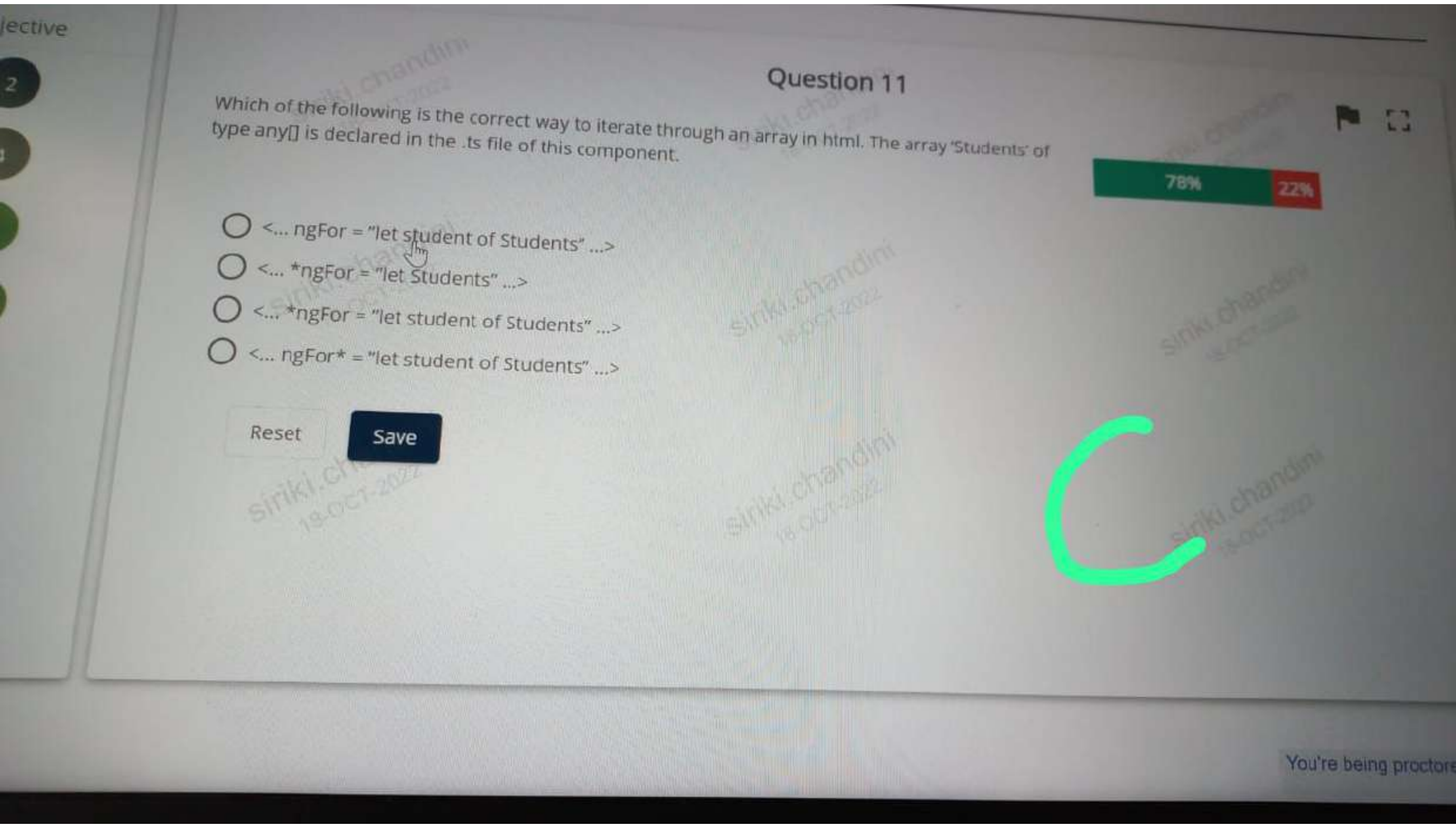
○ < input name="email" #emailRef="ngModel" ngModel required="^[\w-\.]+@([\w-]+\.)">

○ < input type="email" name="email" #emailRef="ngModel" ngModel pattern="^[\w-\.]+@([\w-]+\.)+[\w-\.] +[\w-]{2,4}">

○ < input type="email" name="email" #emailRef="ngModel" ngModel required pattern="^[\w-\.]+([\w-]+\.)+[\w-]{2,4}$">

○ < input type="email" name="email" #emailRef="ngModel" ngModel required pattern="^[\w-\.]+@([\w-]+\.)+[\w-]{2,4}$">

d



Question 11

Which of the following is the correct way to iterate through an array in html. The array 'Students' of type any[] is declared in the .ts file of this component.

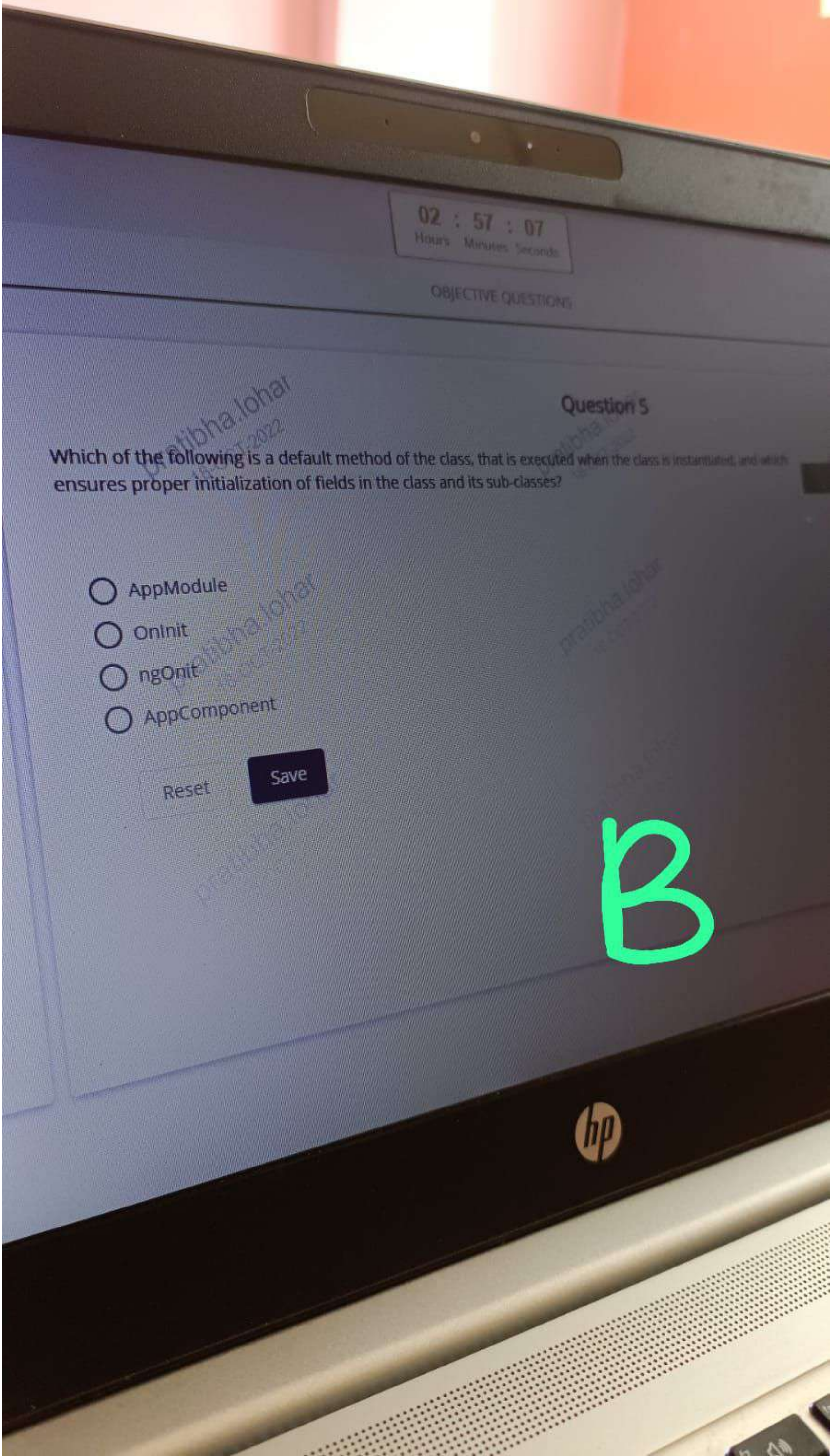
○ <... ngFor = "let student of Students" ...>

○ <... *ngFor = "let Students" ...>

○ <... *ngFor = "let student of Students" ...>

○ <... ngFor* = "let student of Students" ...>

C



02 : 57 : 07
Hours Minutes Seconds

OBJECTIVE QUESTIONS

Question 5

Which of the following is a default method of the class, that is executed when the class is instantiated, and which ensures proper initialization of fields in the class and its sub-classes?

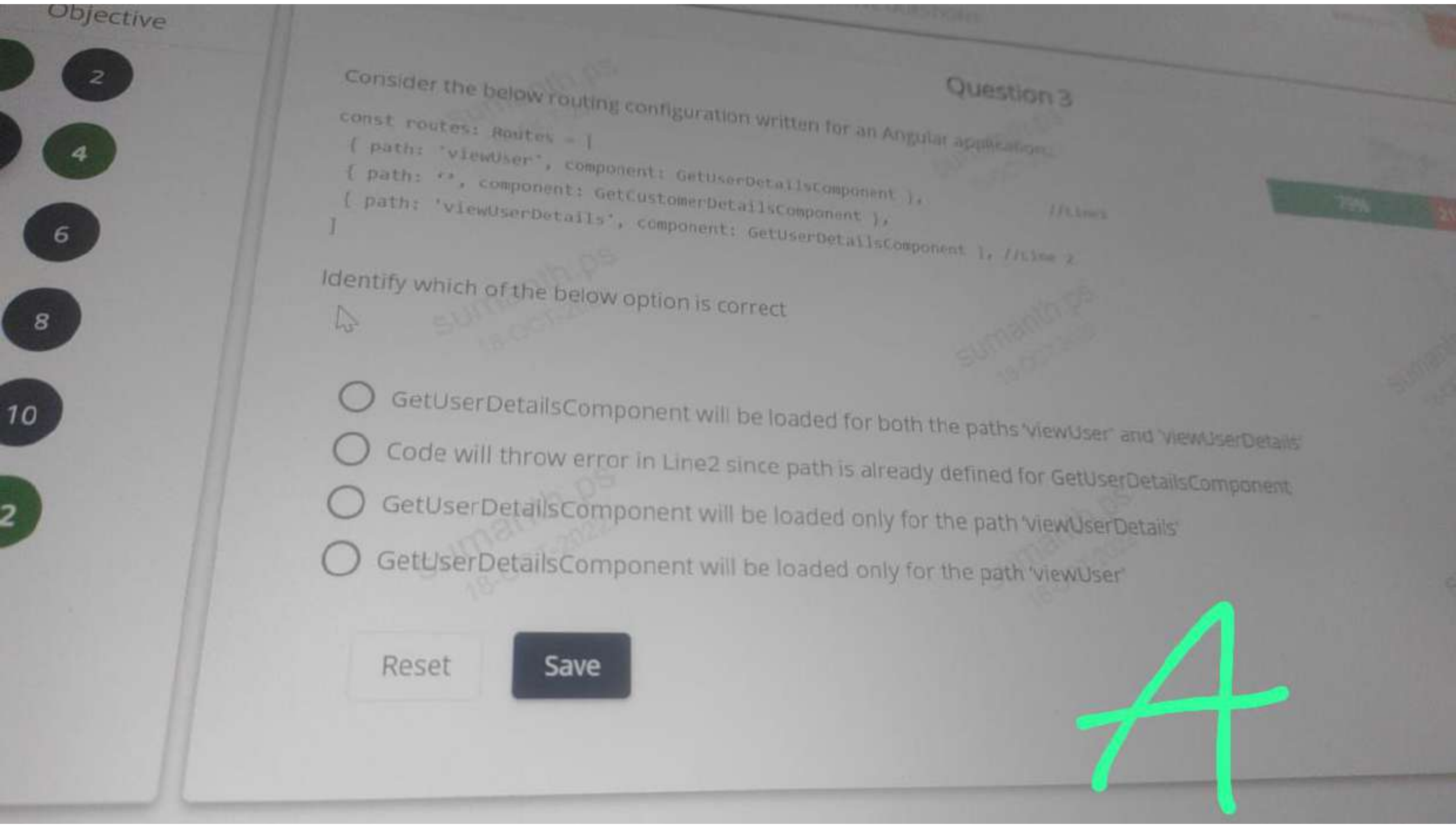
○ AppModule

○ OnInit

○ ngOnit

○ AppComponent

B



Question 3

Consider the below routing configuration written for an Angular application.

```
const routes: Routes = [
 { path: 'viewUser', component: GetUserDetailsComponent },                //Line1
 { path: '', component: GetCustomerDetailsComponent },
 { path: 'viewUserDetails', component: GetUserDetailsComponent }, //Line 2
]
```

Identify which of the below option is correct

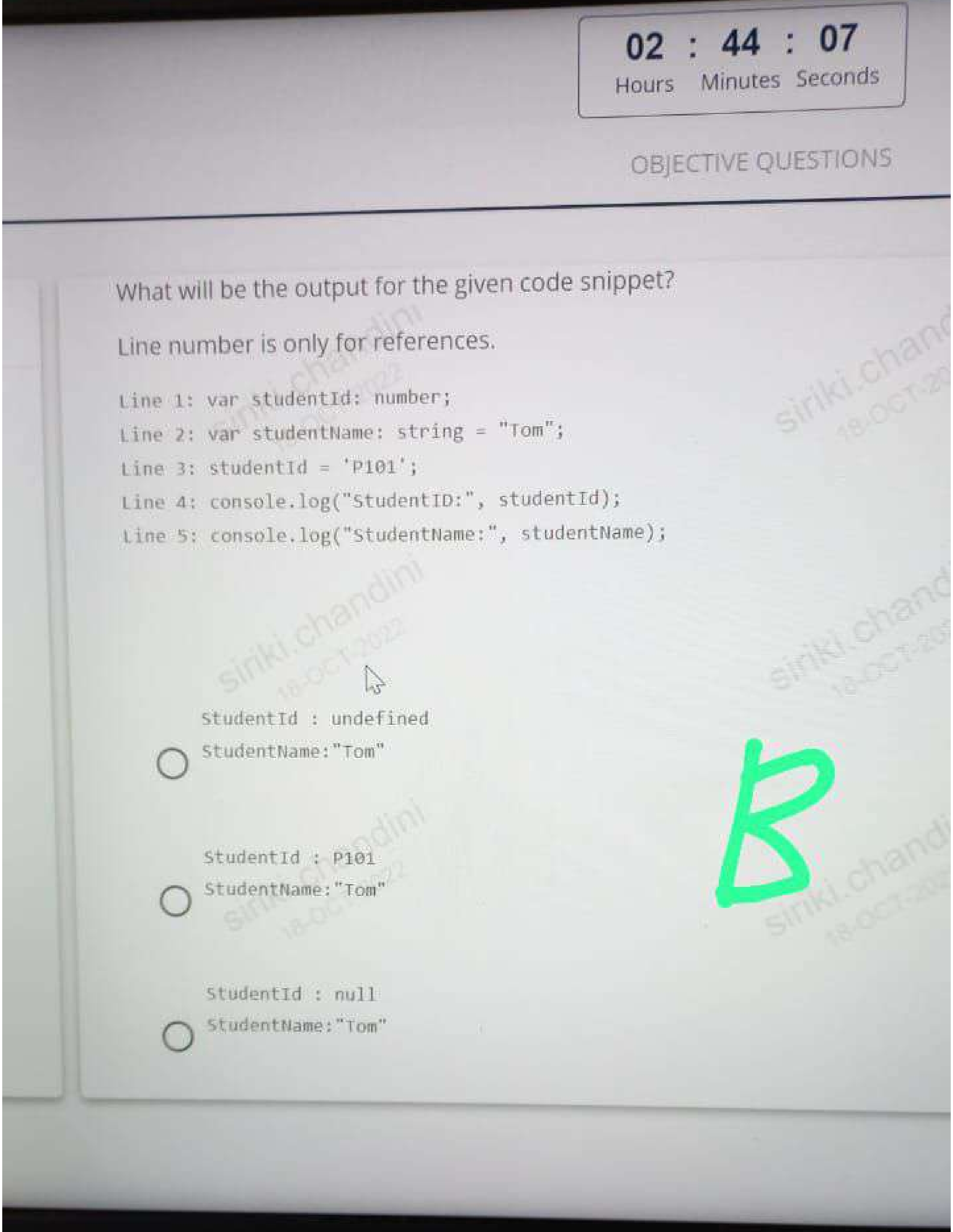
○ GetUserDetailsComponent will be loaded for both the paths 'viewUser' and 'viewUserDetails'

○ Code will throw error in Line2 since path is already defined for GetUserDetailsComponent

○ GetUserDetailsComponent will be loaded only for the path 'viewUserDetails'

○ GetUserDetailsComponent will be loaded only for the path 'viewUser'

A



02 : 44 : 07
Hours Minutes Seconds

OBJECTIVE QUESTIONS

What will be the output for the given code snippet?

Line number is only for references.

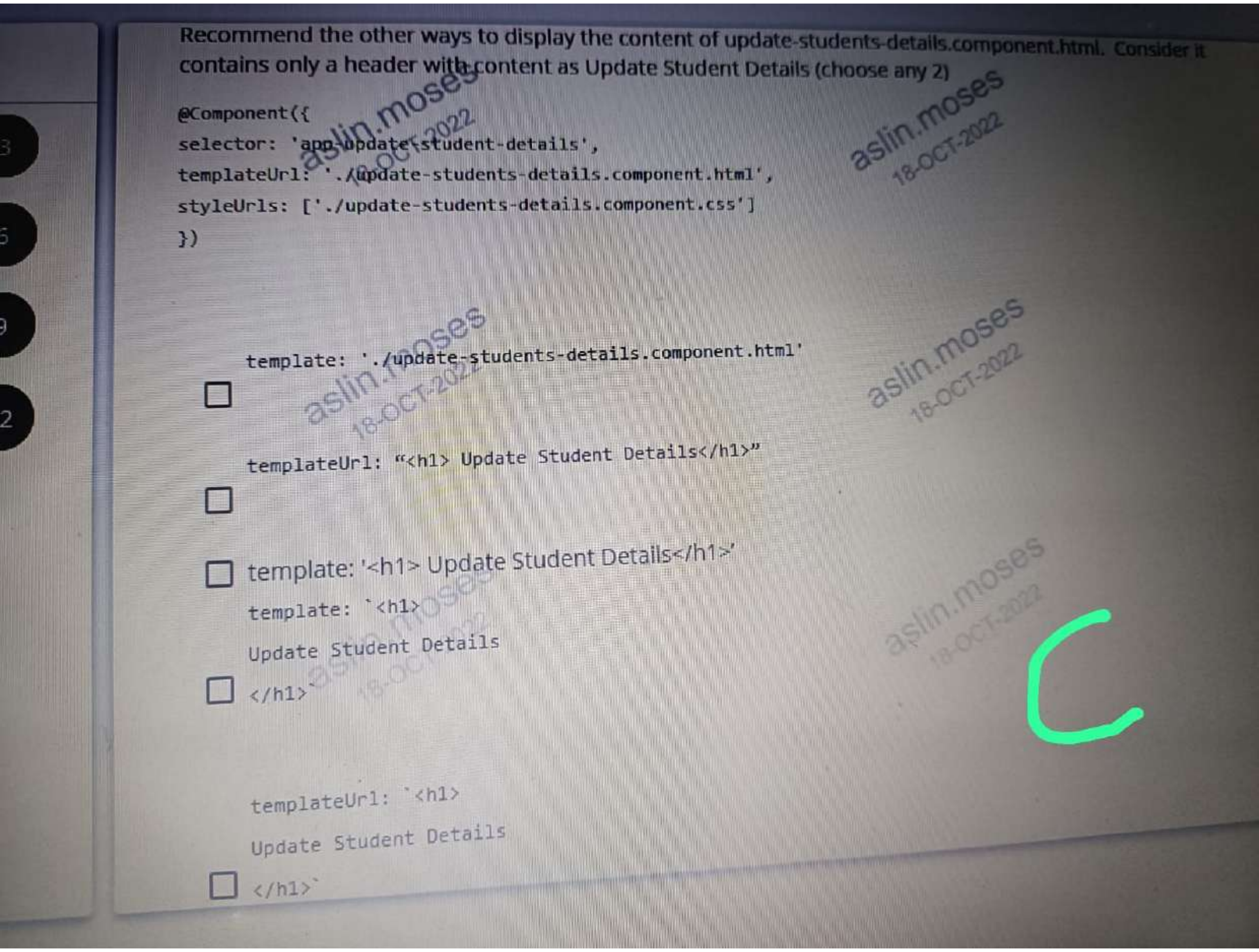
```
Line 1: var studentId: number;
Line 2: var studentName: string = "Tom";
Line 3: studentId = 'P101';
Line 4: console.log("StudentID:", studentId);
Line 5: console.log("StudentName:", studentName);
```

○
```
StudentId : undefined
StudentName:"Tom"
```

○
```
StudentId : P101
StudentName:"Tom"
```

○
```
StudentId : null
StudentName:"Tom"
```

B



Recommend the other ways to display the content of update-students-details.component.html. Consider it contains only a header with content as Update Student Details (choose any 2)

```
@Component({
selector: 'app-update-student-details',
templateUrl: './update-students-details.component.html',
styleUrls: ['./update-students-details.component.css']
})
```

□ template: './update-students-details.component.html'

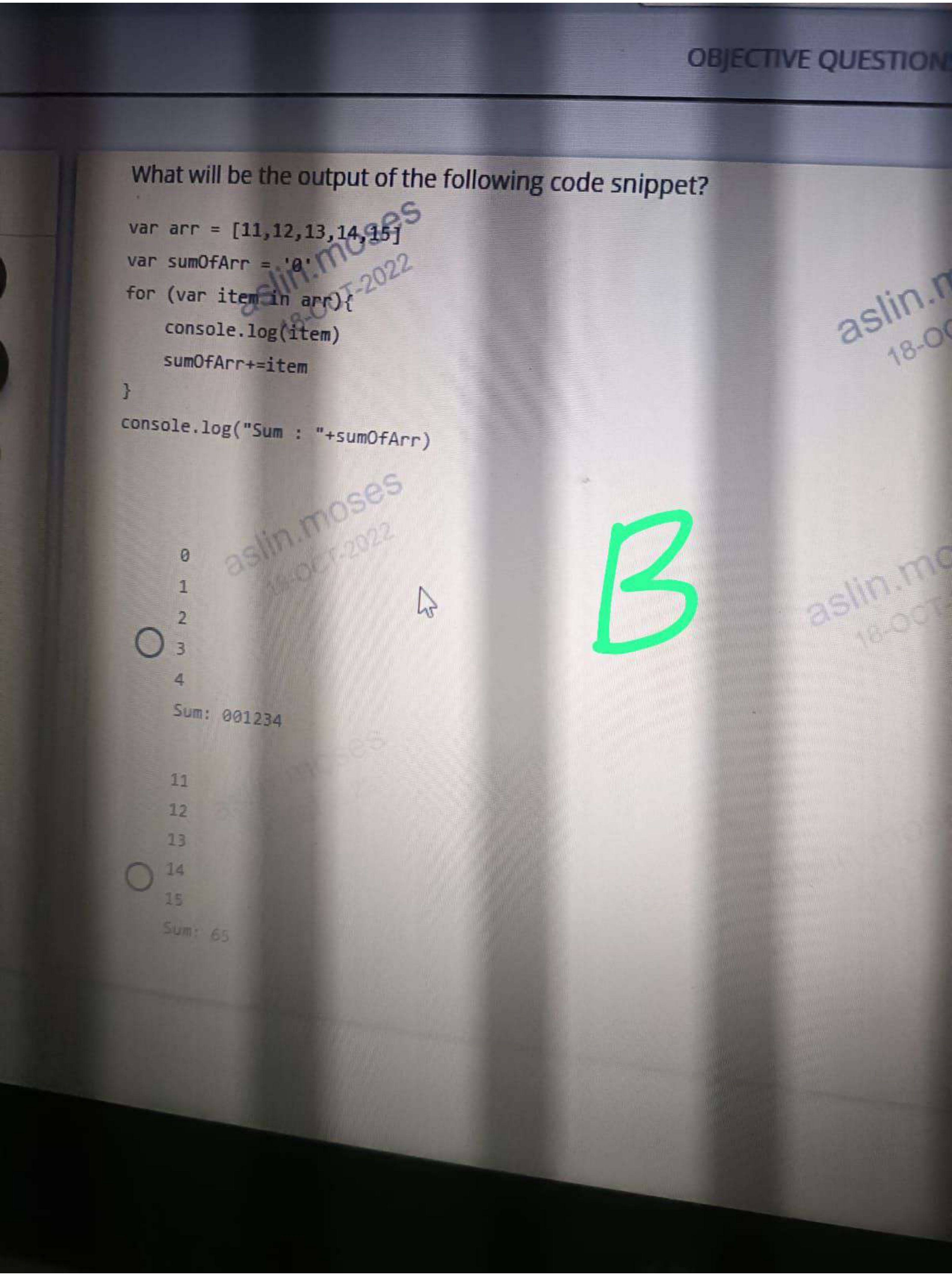
□ templateUrl: “<h1> Update Student Details</h1>”

□ template: '<h1> Update Student Details</h1>'

□ template: `<h1>
Update Student Details
</h1>`

□ templateUrl: `<h1>
Update Student Details
</h1>`

C



OBJECTIVE QUESTION

What will be the output of the following code snippet?

```
var arr = [11,12,13,14,15]
var sumOfArr = '0'
for (var item in arr){
    console.log(item)
    sumOfArr+=item
}
console.log("Sum : "+sumOfArr)
```

B

○
```
0
1
2
3
4
Sum: 001234
```

○
```
11
12
13
14
15
Sum: 65
```