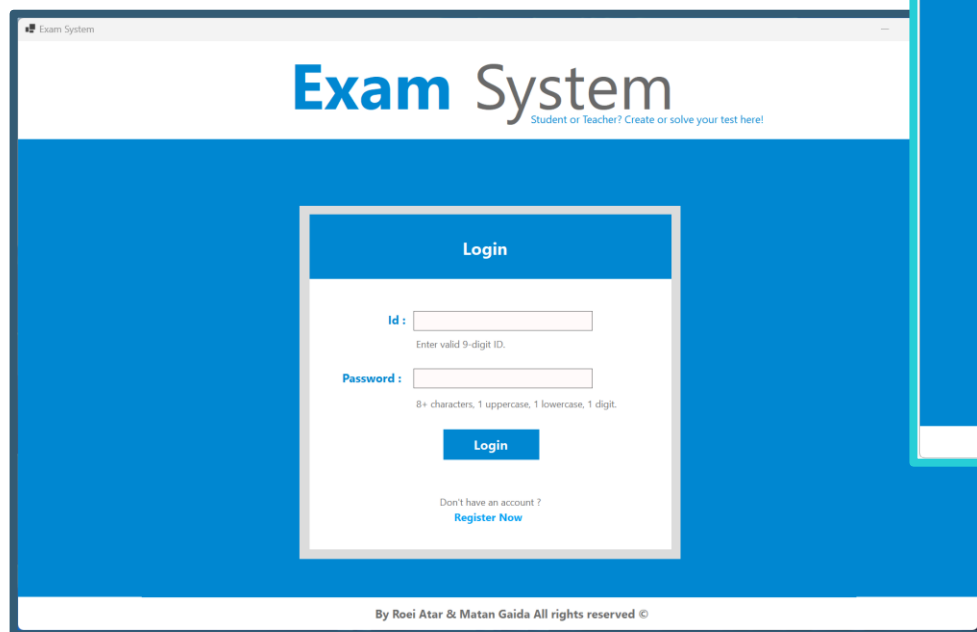


Exam System

By Roei Atar & Matan Gaida

All rights reserved ©



The screenshot shows the login interface of the Exam System. It features a blue header with the title "Exam System" and a subtitle "Student or Teacher? Create or solve your test here!". Below the header is a white box with a blue "Login" title. Inside this box, there are input fields for "Id" and "Password". The "Id" field has a placeholder text "Enter valid 9-digit ID.". The "Password" field has a placeholder text "8+ characters, 1 uppercase, 1 lowercase, 1 digit.". Below the password field is a blue "Login" button. At the bottom of the white box, there is a link "Don't have an account? Register Now". The footer of the page says "By Roei Atar & Matan Gaida All rights reserved ©".

Exam System

Student or Teacher? Create or solve your test here!

Login

Id :

Enter valid 9-digit ID.

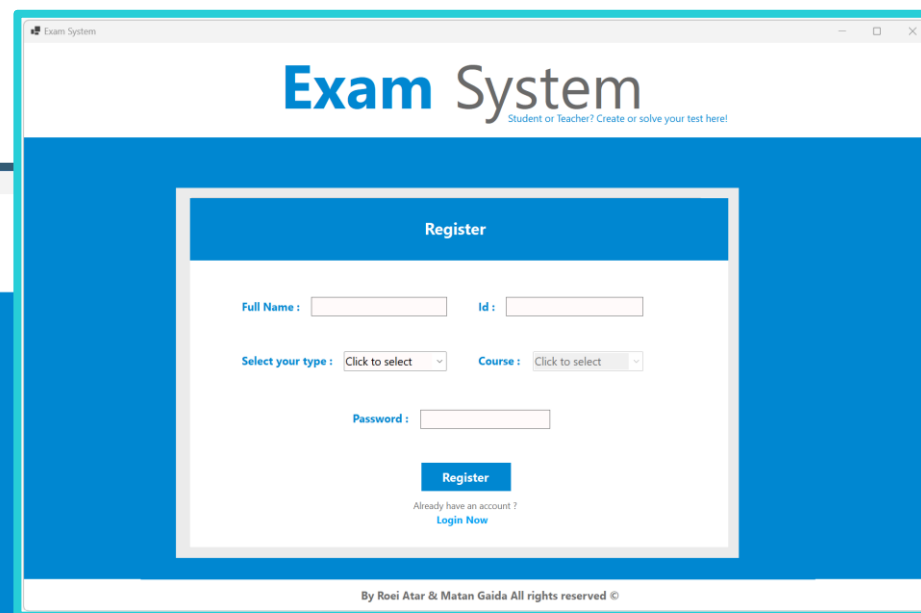
Password :

8+ characters, 1 uppercase, 1 lowercase, 1 digit.

Login

Don't have an account ?
[Register Now](#)

By Roei Atar & Matan Gaida All rights reserved ©



The screenshot shows the registration interface of the Exam System. It features a blue header with the title "Exam System" and a subtitle "Student or Teacher? Create or solve your test here!". Below the header is a white box with a blue "Register" title. Inside this box, there are input fields for "Full Name", "Id", "Select your type", "Course", and "Password". The "Select your type" and "Course" fields are dropdown menus with the placeholder text "Click to select". Below the password field is a blue "Register" button. At the bottom of the white box, there is a link "Already have an account? Login Now". The footer of the page says "By Roei Atar & Matan Gaida All rights reserved ©".

Exam System

Student or Teacher? Create or solve your test here!

Register

Full Name :

Id :

Select your type :

Course :

Password :

Register

Already have an account ?
[Login Now](#)

By Roei Atar & Matan Gaida All rights reserved ©

About

- ▶ This project is about managing a test interface for both teachers and students.
- ▶ For each of the users, the application provides different options with the help of different management screens.
- ▶ The Teacher has the options of creating, saving, editing, updating and deleting tests within a database.
- ▶ In addition, he has the option to see statistical data for the test he has chosen.
- ▶ Student can search and solve an exam And see result of finished exams.

General structure ▶ Client Side

Windows form application .NET framework 6.0

▶ Forms :

- BuildExamForm.cs
- ExamResultForm.cs
- ExamRunForm.cs
- StatisticsForm.cs
- StudentForm.cs
- TeacherForm.cs

▶ Models :

- Error.cs
- Exam.cs
- OptionAns.cs
- Participation.cs
- Question.cs
- User.cs

▶ FormModels :

- ExamFormModel.cs
- RunExamFormModel.cs
- StudentFormModel.cs
- TeacherFormModel.cs

▶ HttpRequestor :

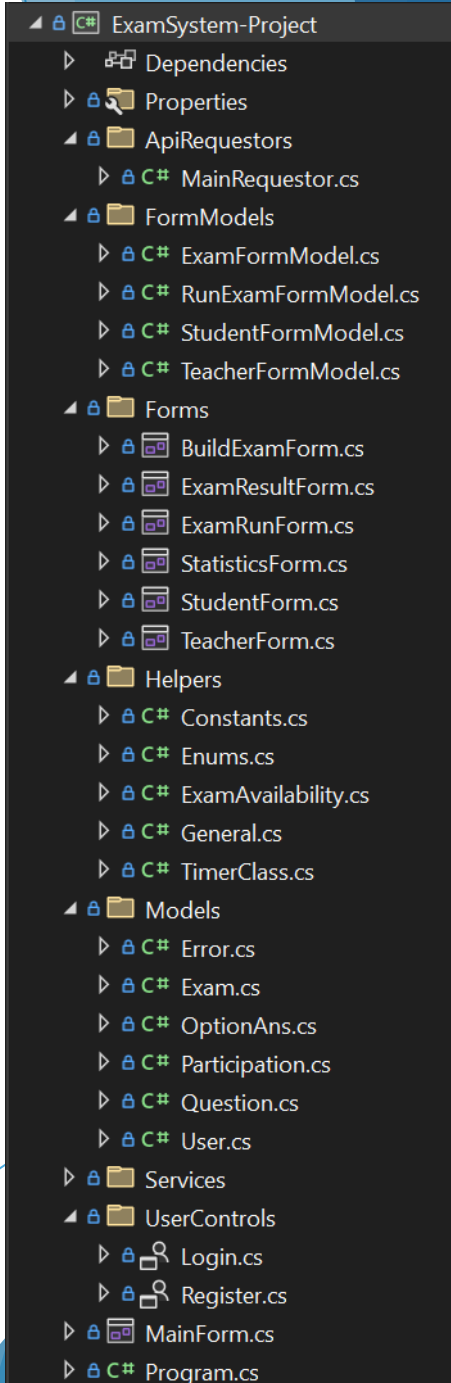
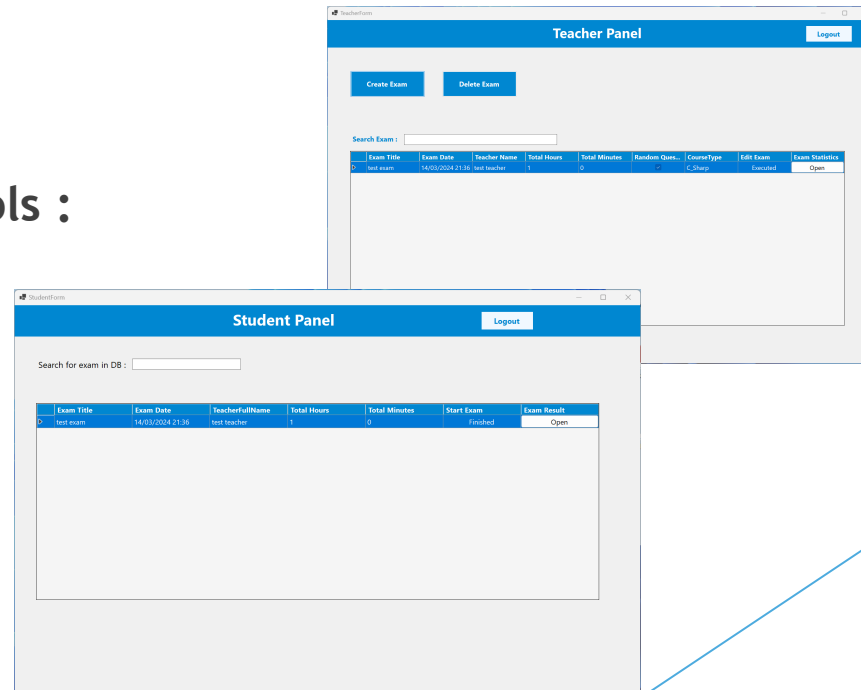
- MainRequestor.cs

▶ Helpers :

- Constants.cs
- Enums.cs
- ExamAvailability.cs
- General.cs
- TimerClass.cs

▶ UserControls :

- Login.cs
- Register.cs



General structure

Server Side

ASP .NET WEB API SERVER

Models :

- Error.cs
- Exam.cs
- OptionAns.cs
- Participation.cs
- Question.cs
- User.cs

DbModels :

- AppDbContext.cs

Controllers :

- ExamsController.cs
- ParticipationsController.cs
- UsersController.cs

Repositories :

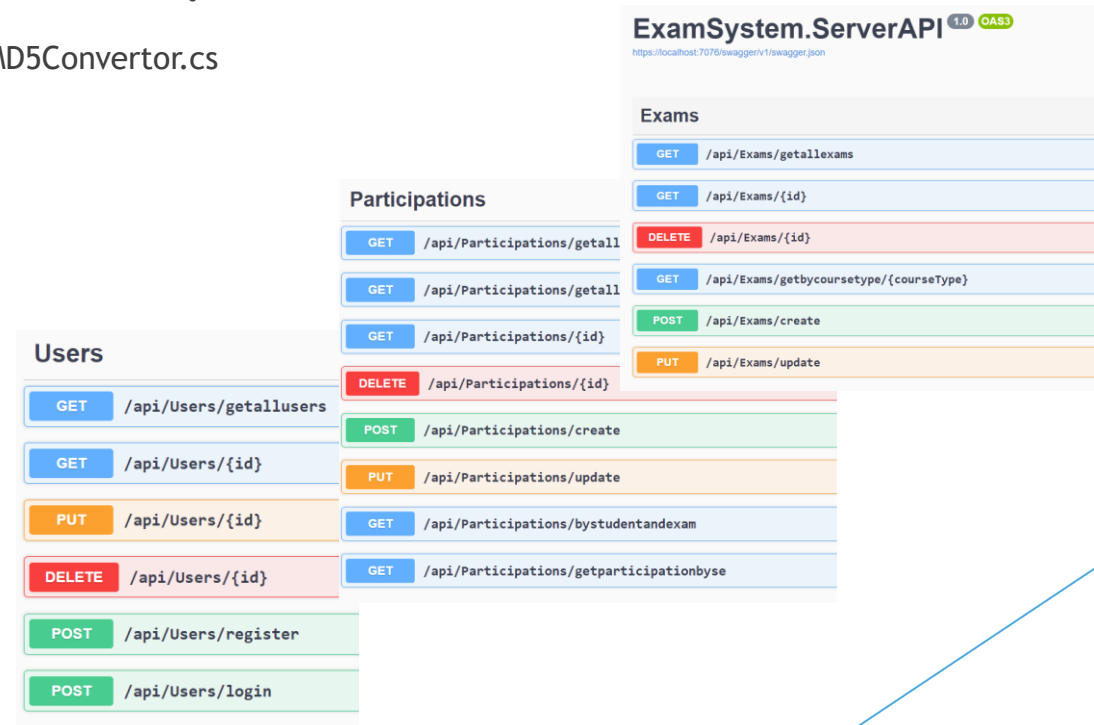
- SystemRepository.cs

Interfaces :

- ISystemRepository.cs

Services :

- MD5Convertor.cs



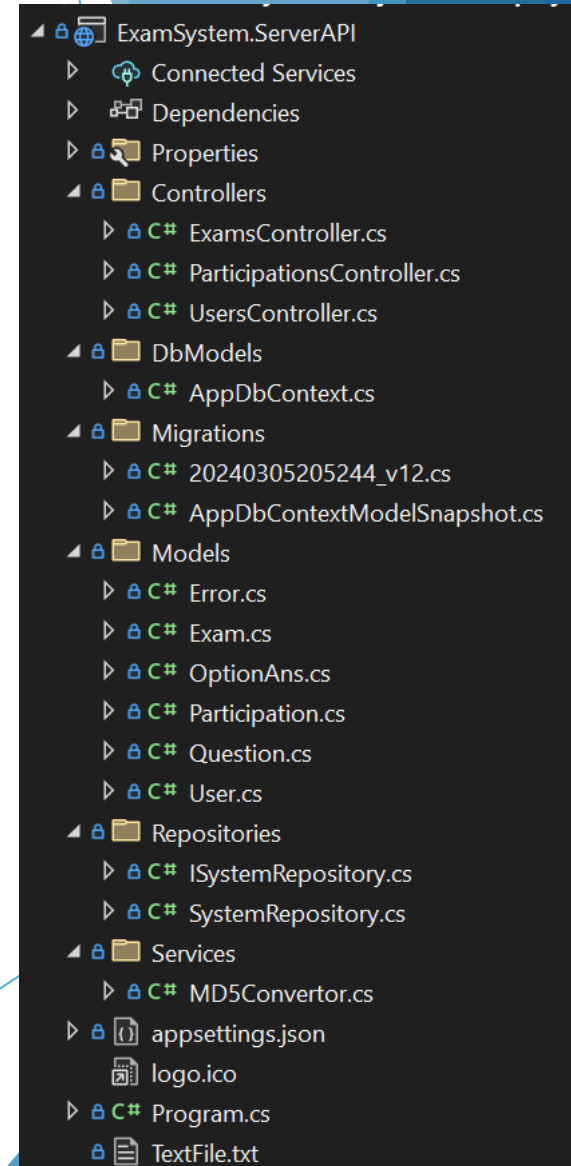
ExamSystem.ServerAPI 1.0 OAS3

<https://localhost:7076/swagger/v1/swagger.json>

Users	
GET	/api/Users/getallusers
GET	/api/Users/{id}
PUT	/api/Users/{id}
DELETE	/api/Users/{id}
POST	/api/Users/register
POST	/api/Users/login

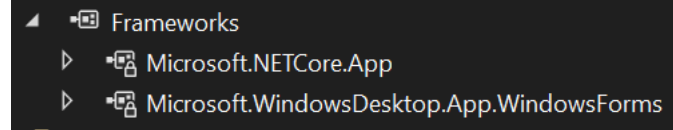
Participations	
GET	/api/Participations/getall
GET	/api/Participations/getall
GET	/api/Participations/{id}
DELETE	/api/Participations/{id}
POST	/api/Participations/create
PUT	/api/Participations/update
GET	/api/Participations/bystudentandexam
GET	/api/Participations/getparticipationbyse

Exams	
GET	/api/Exams/getallexams
GET	/api/Exams/{id}
DELETE	/api/Exams/{id}
GET	/api/Exams/getbycoursetype/{courseType}
POST	/api/Exams/create
PUT	/api/Exams/update

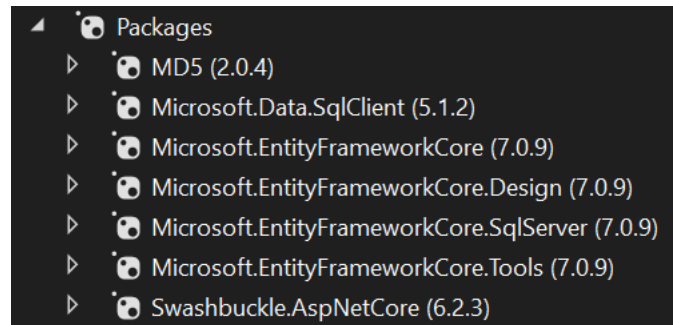
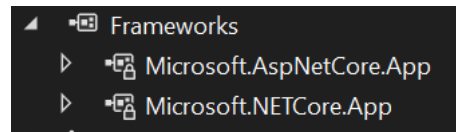


Frameworks & Nuget Technologies

- ▶ Client : For the client side we used the following Frameworks as required:



- ▶ Server : In the server side we used all necessary NuGet packages for accessing and working with a database



List of API functions for accessing server

► Users

```
#region Users
2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<bool> UserRegister(User user)...

2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<User> GetUserById(string userId)...

2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<User> ValidateUser(string password, string userId)...

2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<List<User>> GetAllUsers()...
#endregion
```

► Exams

```
#region Exams
2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<List<Exam>> GetAllExams()...

2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<Exam> GetExamById(int id)...

2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<bool> CreateExam(Exam exam)...

2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<bool> UpdateExam(Exam updatedExam)...

2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<List<Exam>> GetExamsByCourseType(Course_Enum courseType)...

2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<bool> DeleteExam(int id)...
```

► SystemRepository functions

► Participations

```
#region Participations
2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<List<Participation>> GetAllParticipations()...

2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<List<Participation>> GetAllParticipationsById(int examId)...
2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<Participation> GetParticipationById(int id)...

2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<bool> CreateParticipation(Participation participation)...

2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<bool> UpdateParticipation(Participation updatedParticipation)...

2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<bool> DeleteParticipation(int id)...

2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<bool> GetByStudentAndExamId(string studentId, int examId)...

2 references | roeiatar87, 19 hours ago | 1 author, 1 change
public async Task<Participation> GetParticipationBySE(string studentId, int examId)...
#endregion
```

List of API functions for accessing server ► Controllers functions

► Users

```
[HttpGet("getallusers")]
0 references | roeiatar87, 74 days ago | 1 author, 3 changes
public async Task<List<User>> GetAllUsers()...
```

```
[HttpGet("{id}")]
0 references | roeiatar87, 31 days ago | 1 author, 3 changes
public async Task<User> GetUser(string id)...
```

```
[HttpPost("register")]
0 references | roeiatar87, 74 days ago | 1 author, 1 change
public async Task<bool> Post_Register([FromBody] User user)...
```

```
[HttpPost("login")]
0 references | roeiatar87, 74 days ago | 1 author, 1 change
public async Task<User> Post_Login(User user)...
```

```
[HttpPut("{id}")]
0 references | roeiatar87, 58 days ago | 1 author, 2 changes
public void Put(int id, [FromBody] string value)...
```

```
[HttpDelete("{id}")]
0 references | roeiatar87, 58 days ago | 1 author, 2 changes
public void Delete(int id)...
```

► Exams

```
[HttpGet("getallexams")]
0 references | roeiatar87, 58 days ago | 1 author, 1 change
public async Task<List<Exam>> GetAllExams()...
```

```
[HttpGet("{id}")]
0 references | roeiatar87, 58 days ago | 1 author, 1 change
public async Task<Exam> GetExam(int id)...
```

```
[HttpGet("getbycoursetype/{courseType}")]
0 references | matangaida, 8 days ago | 1 author, 1 change
public async Task<List<Exam>> GetExamsByCourseType(Course_Enum courseType)...
```

```
[HttpPost("create")]
0 references | roeiatar87, 58 days ago | 1 author, 1 change
public async Task<bool> CreateExam([FromBody] Exam exam)...
```

```
[HttpPut("update")]
0 references | matangaida, 8 days ago | 2 authors, 2 changes
public async Task<bool> UpdateExam([FromBody] Exam updatedExam)...
```

```
[HttpDelete("{id}")]
0 references | roeiatar87, 58 days ago | 1 author, 1 change
public async Task<bool> DeleteExam(int id)...
```

► Participations

```
[HttpGet("getallparticipations")]
0 references | matangaida, 9 days ago | 1 author, 2 changes
public async Task<List<Participation>> GetAllParticipations()...
```

```
[HttpGet("getallparticipationsbyexamid/{id}")]
0 references | matangaida, 2 days ago | 1 author, 1 change
public async Task<List<Participation>> GetAllParticipationsById(int id)...
```

```
[HttpGet("{id}")]
0 references | matangaida, 2 days ago | 1 author, 1 change
public async Task<Participation> GetParticipationById(int id)...
```

```
[HttpPost("create")]
0 references | matangaida, 9 days ago | 1 author, 2 changes
public async Task<bool> CreateParticipation([FromBody] Participation participation)...
```

```
[HttpPut("update")]
0 references | matangaida, 9 days ago | 1 author, 2 changes
public async Task<bool> UpdateParticipation([FromBody] Participation updatedParticipation)...
```

```
[HttpDelete("{id}")]
0 references | matangaida, 9 days ago | 1 author, 2 changes
public async Task<bool> DeleteParticipation(int id)...
```

```
[HttpGet("bystudentandexam")]
0 references | matangaida, 8 days ago | 1 author, 1 change
public async Task<bool> GetByStudentAndExamId(string studentId, int examId)...
```

```
[HttpGet("getparticipationbyse")]
0 references | matangaida, 2 days ago | 1 author, 1 change
public async Task<Participation> GetParticipationBySE(string studentId, int examId)...
```

Tables definitions in the AppDbContext

```
7 references | roeiatar87, 58 days ago | 1 author, 1 change  
public DbSet<Exam> Exams { get; set; }  
1 reference | roeiatar87, 58 days ago | 1 author, 1 change  
public DbSet<Question> Questions { get; set; }  
1 reference | roeiatar87, 58 days ago | 1 author, 1 change  
public DbSet<OptionAns> OptionAns { get; set; }  
4 references | roeiatar87, 58 days ago | 1 author, 1 change  
public DbSet<User> Users { get; set; }  
3 references | roeiatar87, 18 days ago | 1 author, 1 change  
public DbSet<Error> Errors { get; set; }  
9 references | roeiatar87, 18 days ago | 1 author, 1 change  
public DbSet<Participation> Participations { get; set; }
```

In addition to these definitions, there is also the definition of the relationships between the tables:

- one Exam has -many Questions.
- one Question has - many Answers.
- one Exam has - many Participations.
- one Participation has - many Errors.

Tables and columns in the database

► Users

- Id (PK, int, not null)
- TypeOfUser (nvarchar(max), null)
- FullName (nvarchar(max), null)
- PassWord (nvarchar(max), not null)
- UserId (nvarchar(max), not null)
- CourseType (int, null)

► Exams

- ExamId (PK, int, not null)
- ExamStrId (uniqueidentifier, not null)
- ExamTitle (nvarchar(max), not null)
- ExamDateTime (datetime2(7), not null)
- TeacherFullName (nvarchar(max), not null)
- TotalHours (int, not null)
- TotalMinutes (int, not null)
- RandomQuestionOrder (bit, not null)
- CourseType (int, not null)

► Questions

- QuestionId (PK, int, not null)
- QuestionStrId (uniqueidentifier, not null)
- Text (nvarchar(max), not null)
- ExamStrId (uniqueidentifier, not null)
- RandomAnsOrder (bit, not null)
- IndexCorrect (int, not null)
- ExamId (FK, int, null)

► OptionAns

- OptionAnsId (PK, int, not null)
- OptionAnsStrId (uniqueidentifier, not null)
- QuestionStrId (uniqueidentifier, not null)
- OptionText (nvarchar(max), not null)
- IsCorrect (bit, not null)
- QuestionId (FK, int, null)

► Participations

- ParticipationId (PK, int, not null)
- Student_Id (nvarchar(max), not null)
- Student_Name (nvarchar(max), not null)
- Exam_Id (int, not null)
- Grade (real, not null)

► Errors

- ErrorId (PK, int, not null)
- QuestionContent (nvarchar(max), not null)
- CorrectAnswer (nvarchar(max), not null)
- StudentAnswer (nvarchar(max), not null)
- ParticipationId (FK, int, not null)

Tables and columns in the database - screenshots

► OptionAns

	OptionAnsId	OptionAnsStrId	QuestionStrId	OptionText	IsCorrect	QuestionId
	1	c0f7041a-5c6c-...	48a686cb-b33...	test option1	False	1
	2	a8a670a9-6ff1-...	48a686cb-b33...	test option2	False	1
	3	64269c01-8501...	48a686cb-b33...	test option3	False	1
	4	a103a832-cf74...	48a686cb-b33...	test option4	False	1
	5	2a014161-8f07...	48a686cb-b33...	test option5	True	1
	6	a81bb8ef-e2d8...	b636f144-6630...	test option1	False	2
	7	bcea92e7-097b...	b636f144-6630...	test option2	False	2
	8	b58b01bf-3e22...	b636f144-6630...	test option3	False	2
	9	11ea7584-e5ad...	b636f144-6630...	test option4	False	2
	10	667286ba-488...	b636f144-6630...	test option5	True	2

► Participations

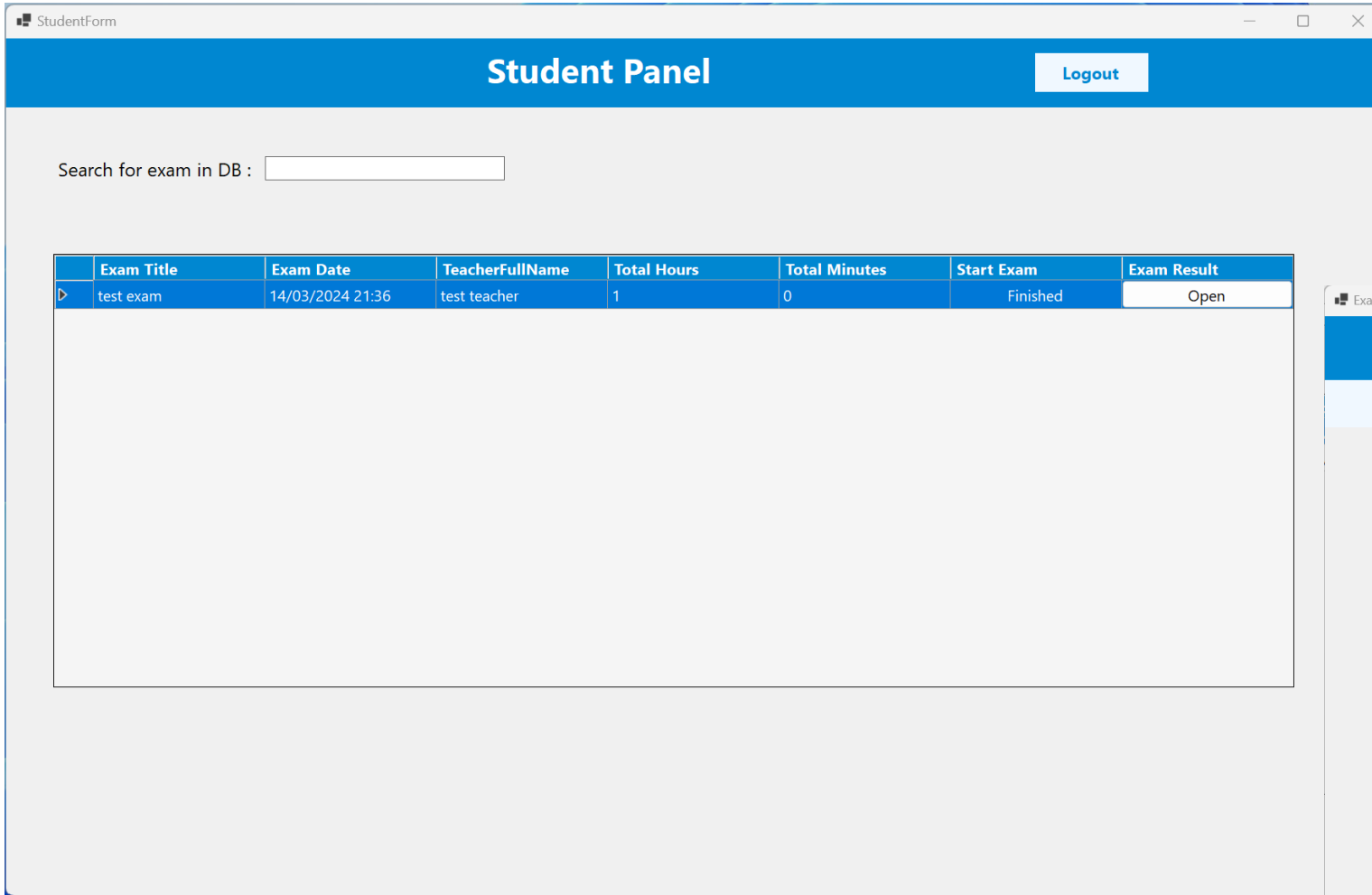
	ParticipationId	Student_Id	Student_Name	Exam_Id	Grade
	7002	300884335	Student Galit	1	67
▶	NULL	NULL	NULL	NULL	NULL

► Errors

	ErrorId	QuestionConte...	CorrectAnswer	StudentAnswer	ParticipationId
	1	what is the best...	test option5	test option1	7002
▶	NULL	NULL	NULL	NULL	NULL

Windows screenshots - Student

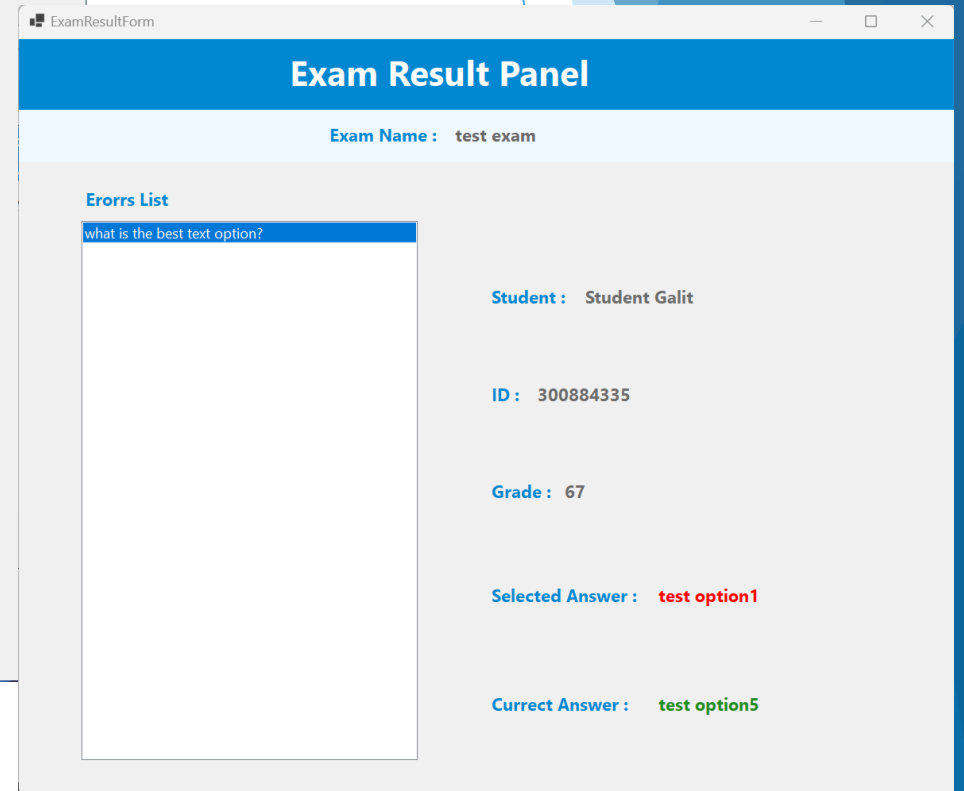
► Student Panel



The screenshot shows a web application window titled "StudentForm". The main header is blue with the text "Student Panel" and a "Logout" button. Below the header, there is a search bar labeled "Search for exam in DB :". Underneath the search bar is a table with 8 columns: Exam Title, Exam Date, TeacherFullName, Total Hours, Total Minutes, Start Exam, and Exam Result. The table contains one row with the following data: "test exam", "14/03/2024 21:36", "test teacher", "1", "0", "Finished", and "Open". Below the table is a large empty rectangular box.

Exam Title	Exam Date	TeacherFullName	Total Hours	Total Minutes	Start Exam	Exam Result
test exam	14/03/2024 21:36	test teacher	1	0	Finished	Open

► Exam Result



The screenshot shows a web application window titled "ExamResultForm". The main header is blue with the text "Exam Result Panel". Below the header, there is a label "Exam Name : test exam". Underneath this is a section titled "Errors List" which contains a text input field with the value "what is the best text option?". To the right of the input field, there are several labels and values: "Student : Student Galit", "ID : 300884335", "Grade : 67", "Selected Answer : test option1", and "Current Answer : test option5".

Exam Name : test exam

Errors List

what is the best text option?

Student : Student Galit

ID : 300884335

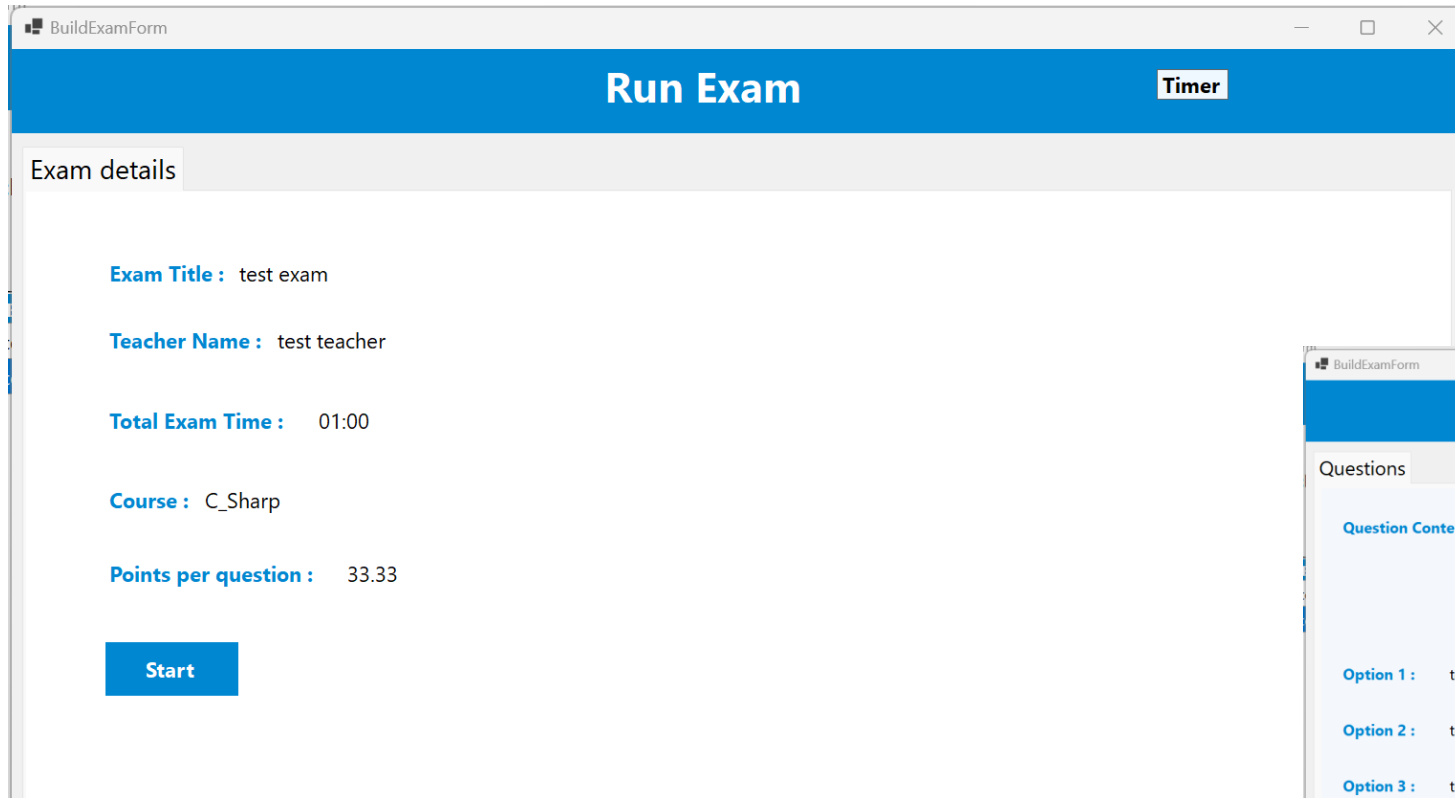
Grade : 67

Selected Answer : test option1

Current Answer : test option5

Windows screenshots - Student

▶ Exam details

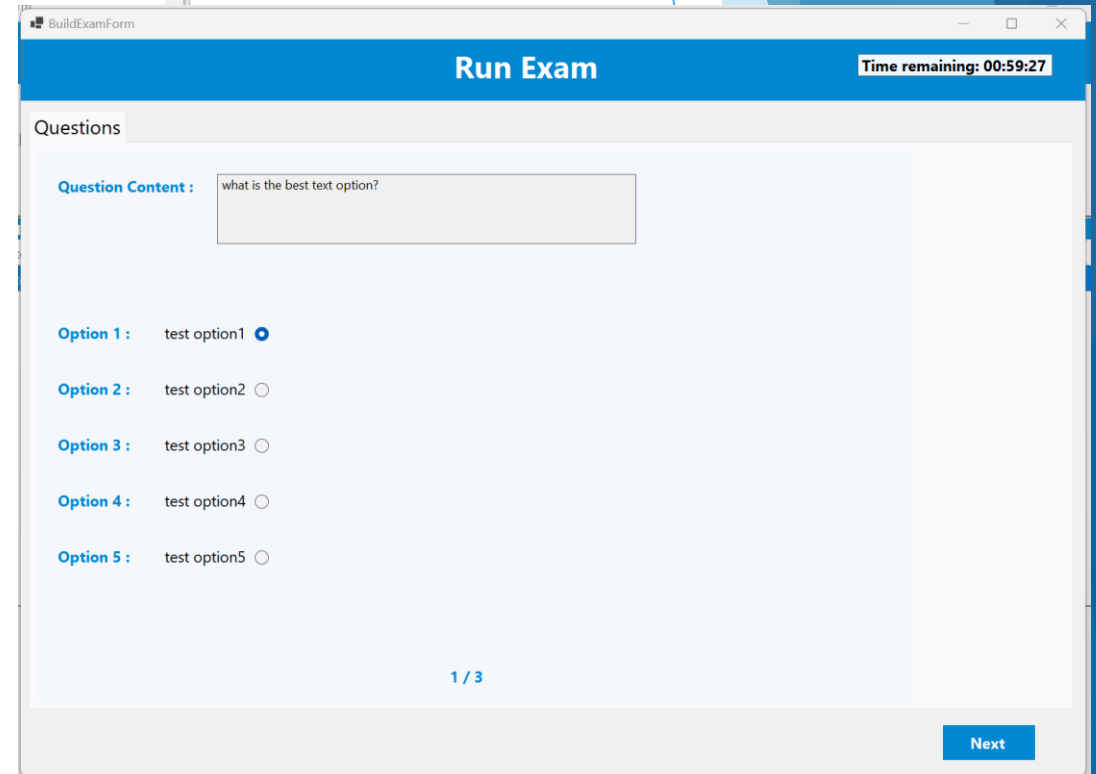


The screenshot shows a window titled 'BuildExamForm' with a blue header bar containing 'Run Exam' and a 'Timer' button. Below the header is a tab labeled 'Exam details'. The main content area displays the following information:

- Exam Title :** test exam
- Teacher Name :** test teacher
- Total Exam Time :** 01:00
- Course :** C_Sharp
- Points per question :** 33.33

At the bottom left, there is a blue 'Start' button.

▶ Questions



The screenshot shows a window titled 'BuildExamForm' with a blue header bar containing 'Run Exam' and a 'Time remaining: 00:59:27' timer. Below the header is a tab labeled 'Questions'. The main content area displays the following information:

Question Content : what is the best text option?

Below the question content, there are five options, each with a radio button:

- Option 1 :** test option1 ☒
- Option 2 :** test option2 ☐
- Option 3 :** test option3 ☐
- Option 4 :** test option4 ☐
- Option 5 :** test option5 ☐

At the bottom center, there is a '1 / 3' indicator. At the bottom right, there is a blue 'Next' button.

Windows screenshots - Teacher

► Teacher Panel

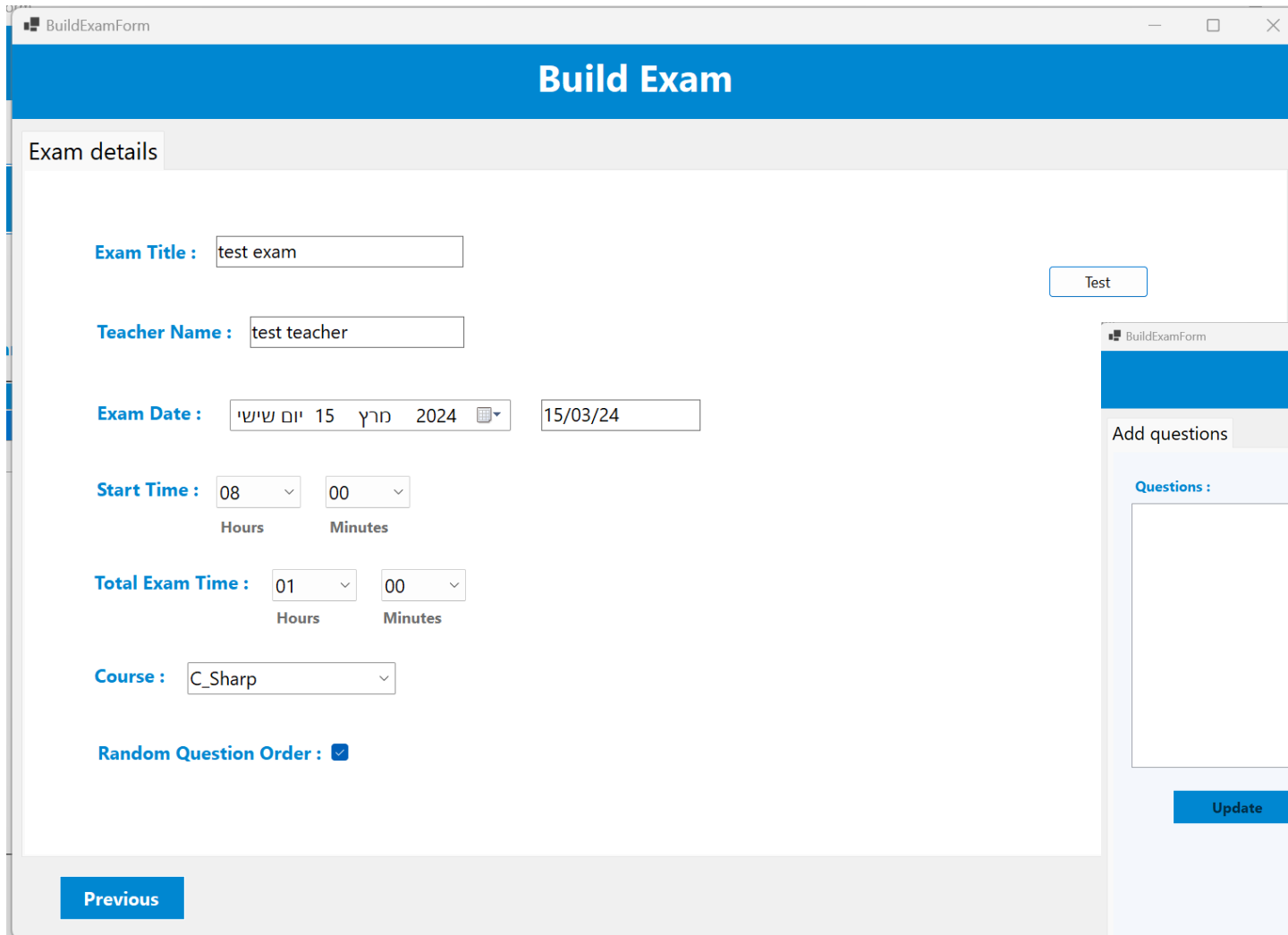
The screenshot shows a web application window titled "TeacherForm". The main header is "Teacher Panel" with a "Logout" button on the right. Below the header, there are two buttons: "Create Exam" and "Delete Exam". A search bar is labeled "Search Exam :". Below the search bar is a table with the following data:

	Exam Title	Exam Date	Teacher Name	Total Hours	Total Minutes	Random Ques...	CourseType	Edit Exam	Exam Statistics
►	test exam	14/03/2024 21:36	test teacher	1	0	✓	C_Sharp	Executed	Open

Below the table is a large empty rectangular area.

Windows screenshots - Teacher

► Build Exam - details

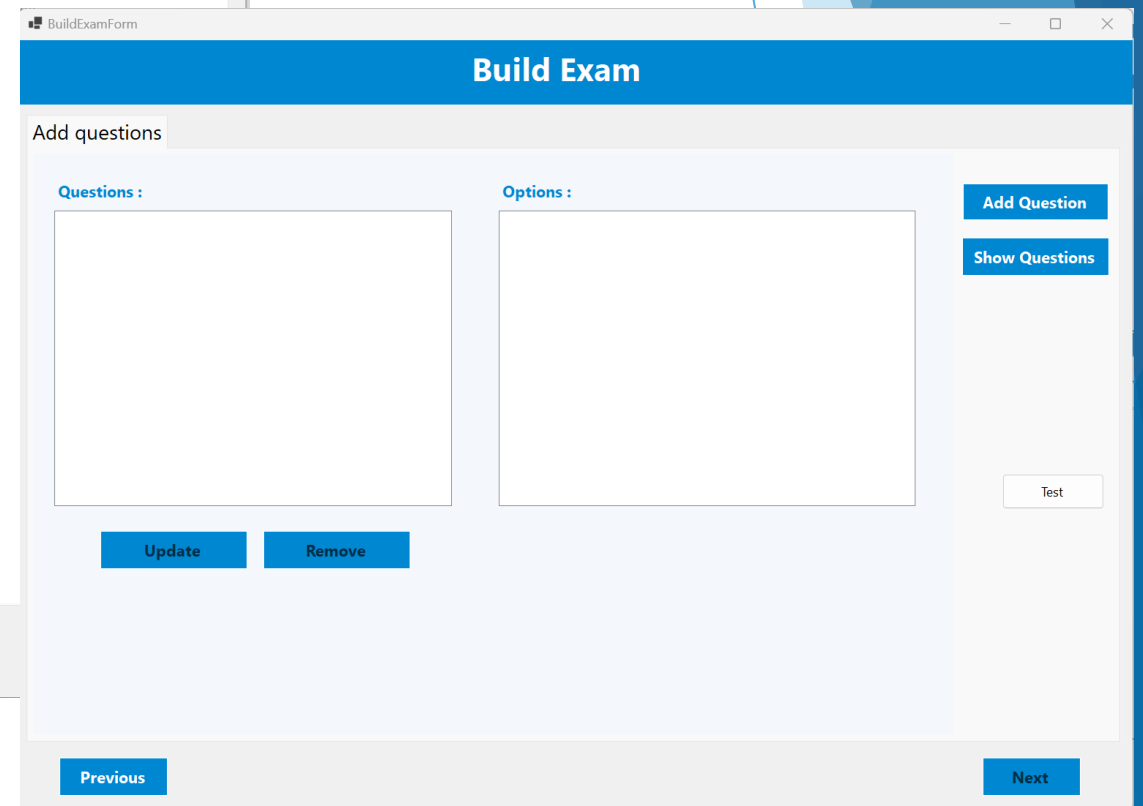


The screenshot shows a web application window titled "BuildExamForm" with a blue header bar labeled "Build Exam". Below the header, there is a tab labeled "Exam details". The form contains several input fields and a checkbox:

- Exam Title :** A text input field containing "test exam".
- Teacher Name :** A text input field containing "test teacher".
- Exam Date :** A date picker showing "יום שישי 15 מרץ 2024" and a text input field containing "15/03/24".
- Start Time :** Two dropdown menus for "Hours" (08) and "Minutes" (00).
- Total Exam Time :** Two dropdown menus for "Hours" (01) and "Minutes" (00).
- Course :** A dropdown menu showing "C_Sharp".
- Random Question Order :** A checked checkbox.

At the bottom left, there is a "Previous" button. At the bottom right, there is a "Test" button.

► Question list view



The screenshot shows the same "BuildExamForm" window, but with the "Add questions" tab selected. The form is divided into two main sections:

- Questions :** A large empty rectangular box for adding questions.
- Options :** A large empty rectangular box for adding options.

At the bottom of the "Questions" section, there are "Update" and "Remove" buttons. At the bottom of the "Options" section, there are "Add Question" and "Show Questions" buttons. At the bottom right, there is a "Test" button. At the bottom left, there is a "Previous" button. At the bottom right, there is a "Next" button.

Windows screenshots - Teacher

► Add Question

The screenshot shows a web application window titled 'BuildExamForm'. The main header is 'Build Exam'. Below it, there's a tab labeled 'Add questions'. The interface is divided into two main sections: 'Question Content' and 'Options'.

Question Content: A text input field contains 'what is the best text option?'. To its right is an 'Add Option' button and a checkbox labeled 'Random Options Order'.

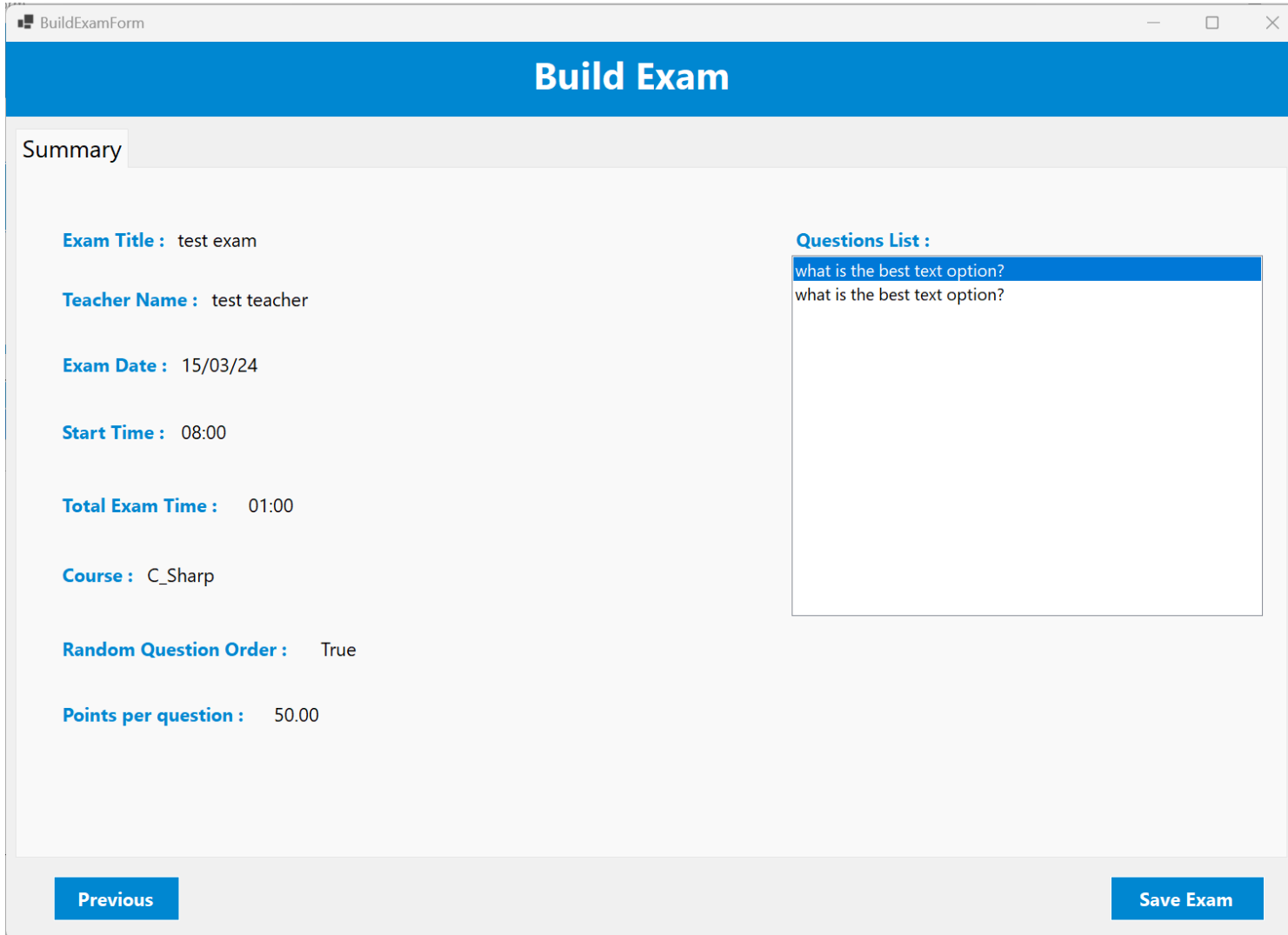
Options: There are five rows, each representing an option. Each row has a label (Option 1 to Option 5), a text input field, a 'Delete' button, and an 'Is Correct' radio button.

Option	Text	Delete	Is Correct
Option 1 :	test option1	Delete	<input checked="" type="radio"/>
Option 2 :	test option2	Delete	<input type="radio"/>
Option 3 :	test option3	Delete	<input type="radio"/>
Option 4 :	test option4	Delete	<input type="radio"/>
Option 5 :	test option5	Delete	<input type="radio"/>

At the bottom right of the options section is a 'Save Question' button. On the right side of the form, there are two buttons: 'Add Question' and 'Show Questions'. At the bottom of the window, there are 'Previous' and 'Next' buttons.

Windows screenshots - Teacher

► Summery



The screenshot shows a Windows application window titled "BuildExamForm". The window has a blue header bar with the text "Build Exam". Below the header, there is a tab labeled "Summary". The main content area is divided into two columns. The left column contains several fields with labels in blue text: "Exam Title : test exam", "Teacher Name : test teacher", "Exam Date : 15/03/24", "Start Time : 08:00", "Total Exam Time : 01:00", "Course : C_Sharp", "Random Question Order : True", and "Points per question : 50.00". The right column has a section titled "Questions List :" in blue text, followed by a list box containing two identical entries: "what is the best text option?". At the bottom of the window, there are two buttons: "Previous" on the left and "Save Exam" on the right.

BuildExamForm

Build Exam

Summary

Exam Title : test exam

Teacher Name : test teacher

Exam Date : 15/03/24

Start Time : 08:00

Total Exam Time : 01:00

Course : C_Sharp

Random Question Order : True

Points per question : 50.00

Questions List :

- what is the best text option?
- what is the best text option?

Previous **Save Exam**

Windows screenshots - Teacher

► Statistics Panel

StatisticForm

Statistics Panel

Exam Name : test exam Students Average : 33.33

Students List

- Student Galit
- Student Lior
- student ayala

Student : Student Galit

ID : 300884335

Grade : 33

Selected Answer : test option2

Correct Answer : test option5

Errors List

- what is the best text option?
- what is the best text option?

Windows screenshots - Teacher

► Statistics Panel

StatisticForm

Statistics Panel

Exam Name : test exam Students Average : 33.33

Students List

- Student Galit
- Student Lior
- student ayala

Student : Student Galit

ID : 300884335

Grade : 33

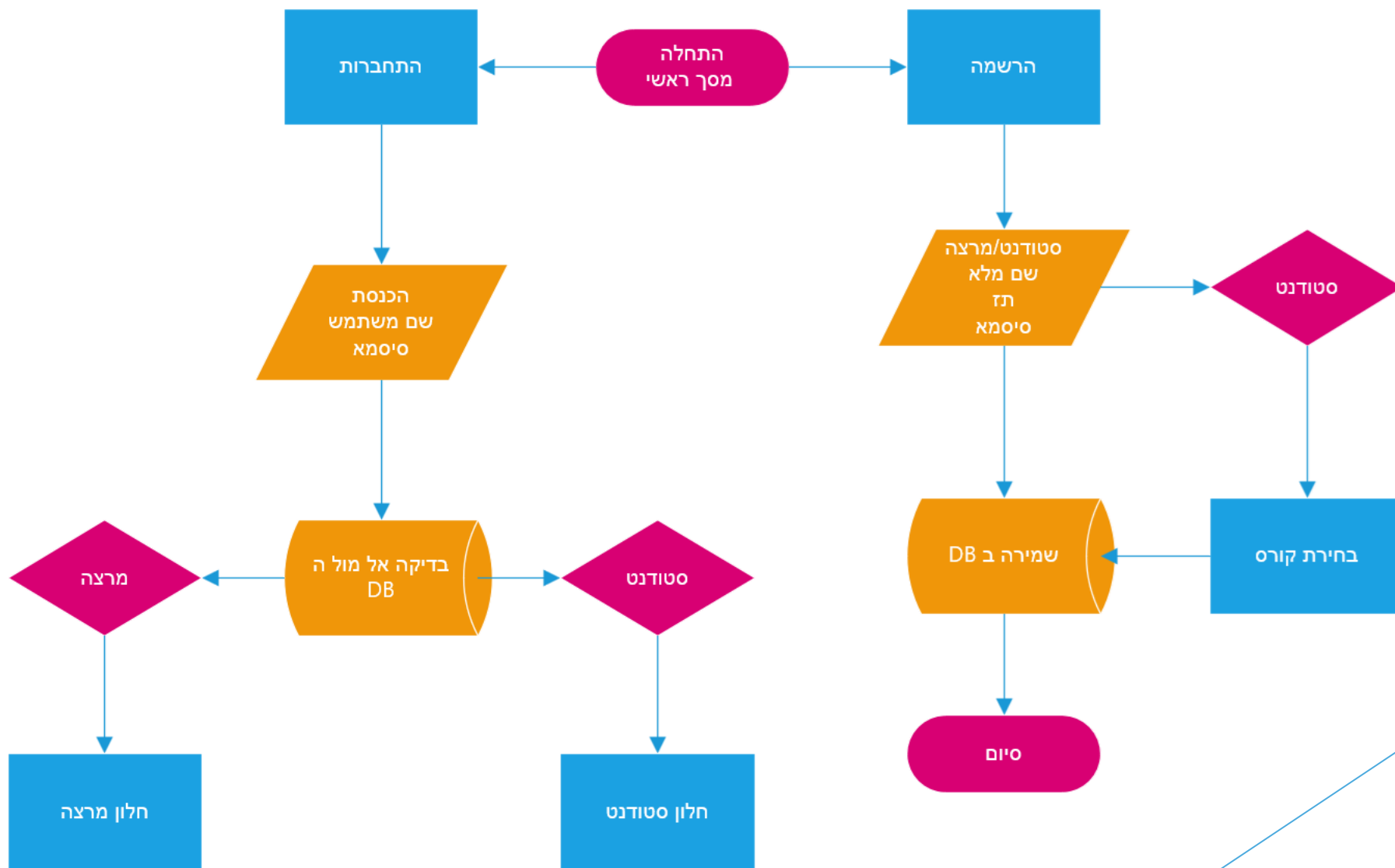
Selected Answer : test option2

Current Answer : test option5

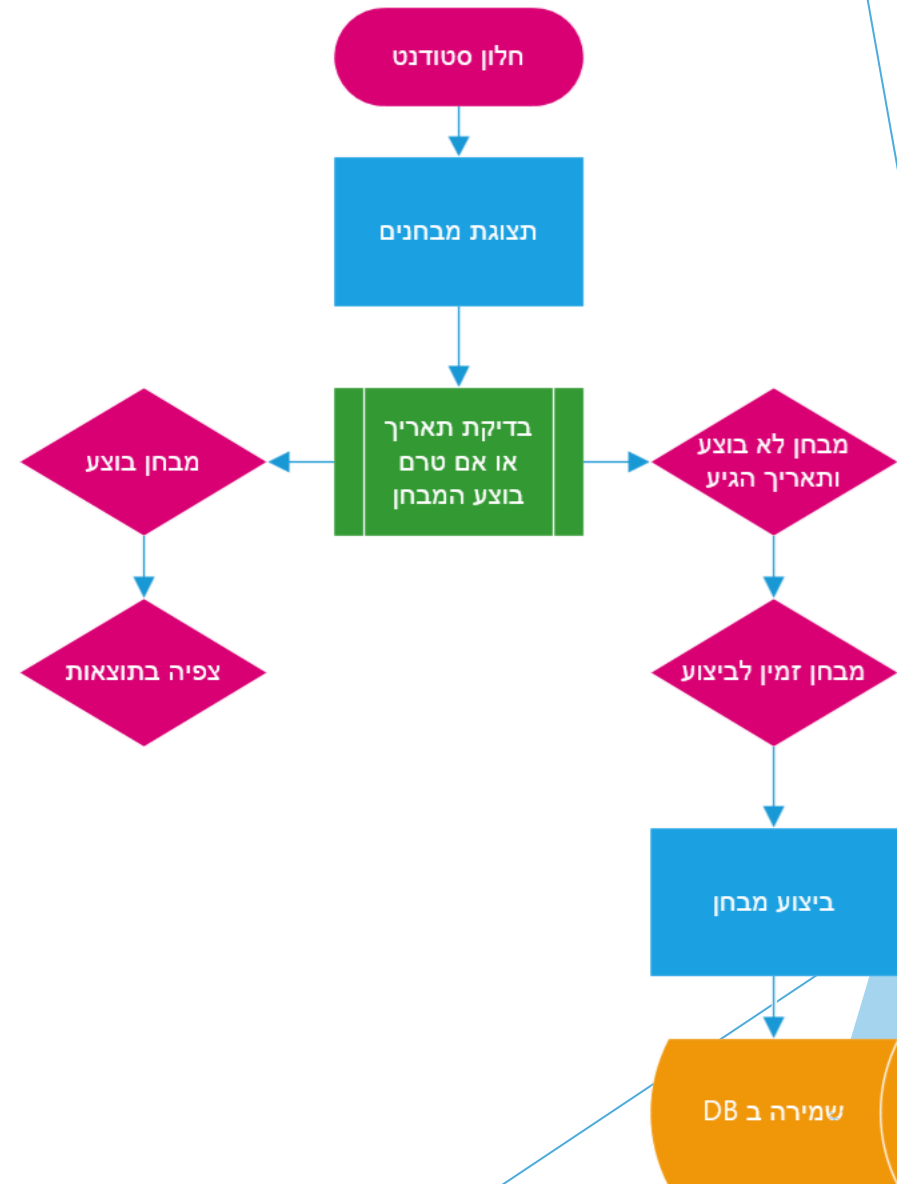
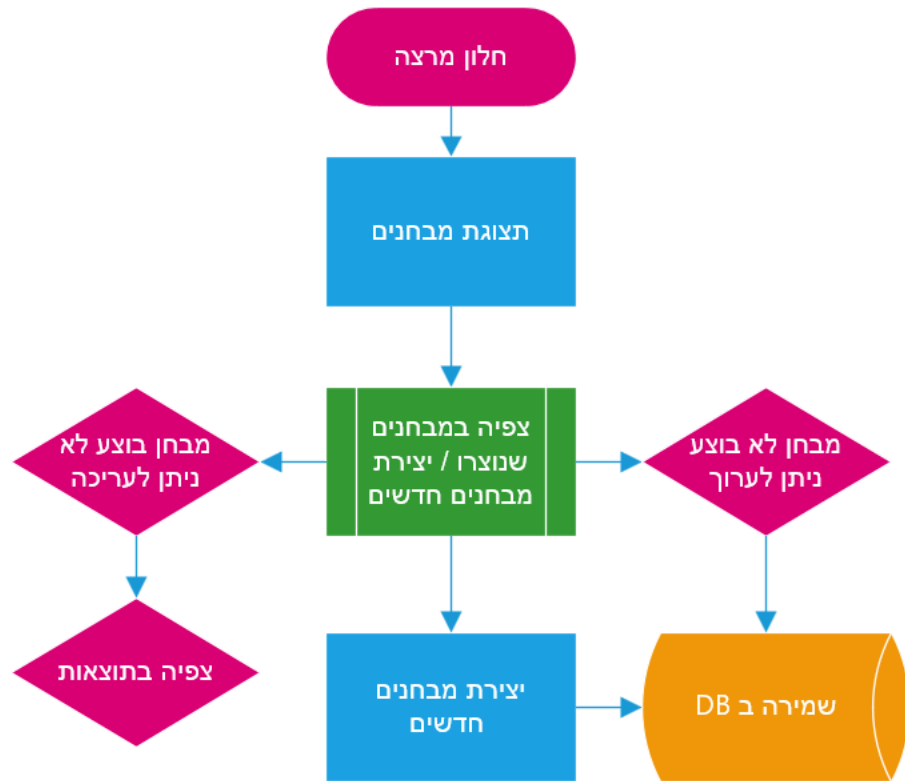
Errors List

- what is the best text option?
- what is the best text option?

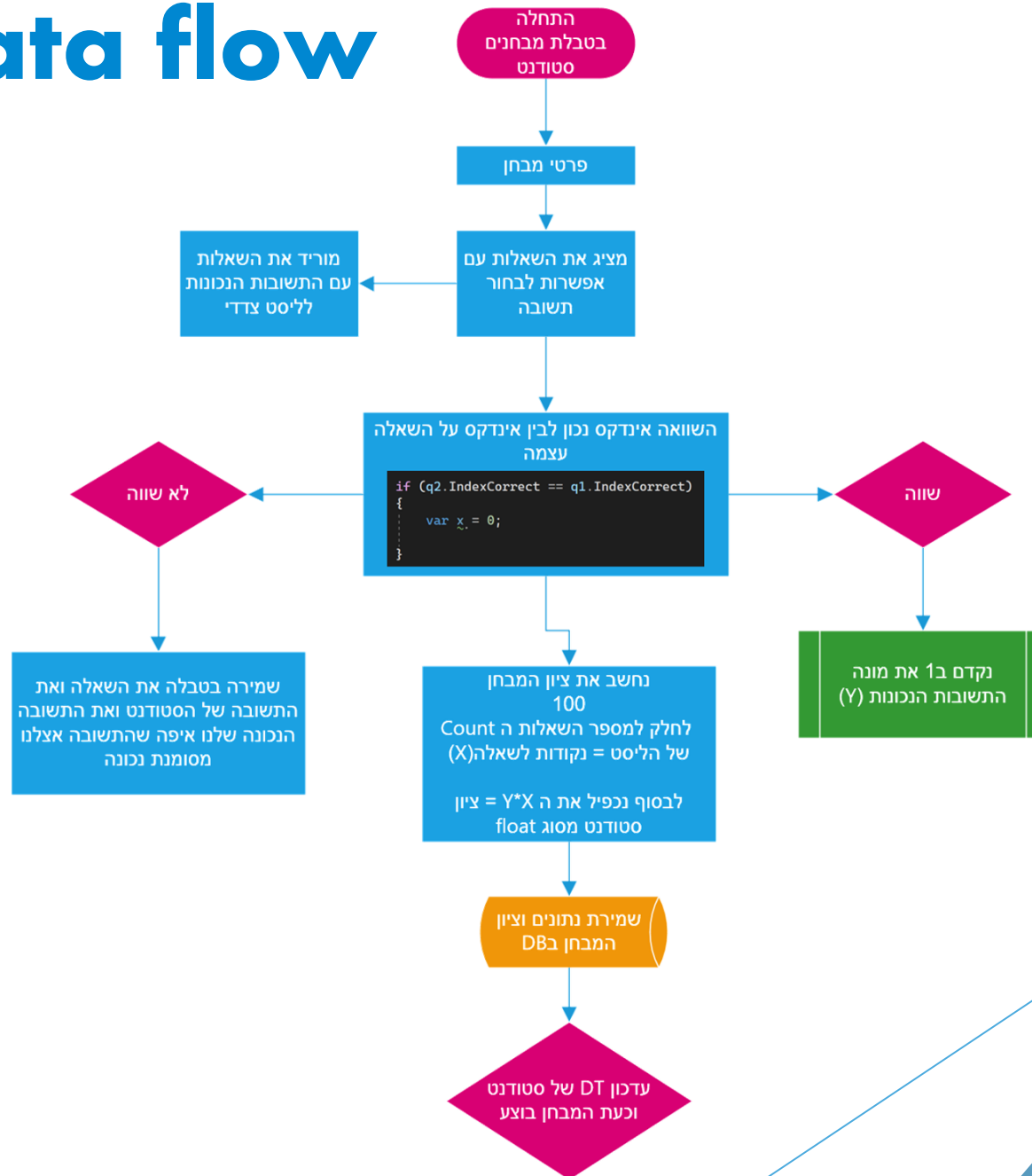
Main data flow



Student & Teacher data flow



Run Exam data flow



THANK YOU

By Roei Atar & Matan Gaida

All rights reserved ©