



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ ΜΑΘΗΜΑΤΟΣ
<<ΕΚΠΑΙΔΕΥΤΙΚΟ ΛΟΓΙΣΜΙΚΟ>>

Γκολέμι Κριστιάν, Α.Μ.: Π18029

Χρήστος Μιχαήλ Καταγής, Α.Μ.: Π18067

Java programming language was used to implement the project and specifically we implemented a Dynamic Web Project using JSP pages and Servlets according to the MVC standard. For this project, we decided to create a web project in which Python Operators would be the main subject for teaching. Python Operators consist of seven subchapters which can be seen by running the application. For the purposes of this project, we implemented theory sections for all the chapters, but evaluation test chapters just for the two first chapters – Arithmetic and Assignment Operators. Implementing the test chapters just for our two first chapters would be enough for the purposes of our project and implementing tests for all the chapters would be an easy to do but extremely time-consuming process, as for each chapter we would have to add totally a minimum of 30 questions manually in the database. The most important folders in which the files we implemented are located are *Java Resources* and *Web Content*. Below we will analyze the files that are in these folders as well as their functionality.

Java Resources

This folder contains the Java source code files of the project and we can see how it consists of three packages: basic, database, servlets.

Package basic

This package contains the files Encryption.java, Functions.java, Student.java. Encryption.java class is responsible for encrypting as well as decrypting the users password registering in the application before it is stored in the database. During the registration process, the password entered by the user is encrypted before being stored in the database, while during the login process, the methods of this file are used again to decrypt the code stored in the database and to check if it is identical to the password entered by the user. Below we can see two photos of the login and registration forms created for our project.



The screenshot shows a web form titled "Educational Software Python Operators". It features two input fields: "E-mail" and "Password". Below these fields is a red "Log In" button. Under the button is a link that says "Register now!". The form is set against a light gray background.

Login form

Educational Software

Enter your data below

E-mail
Password
Confirm Password
First Name
Last Name

Register

Registration form

In the Functions.java file we have implemented methods that are used for various functions of the application. Following we analyze their functionality:

- pickRandomly(): this function is used in order to pick randomly one of the three options given for tests. The options are multiple choice, true or false and gap fill and this function returns a string which represents one of those three options.
- setColor(): this function is used in the test result page of the application. Depending on if the users answer was correct or wrong, this function returns a string which is used in the class of a div element so that it will be colored either in red or in green. Later on there will be further explanation and photos for this specific methods result.
- findSuccessRatio(): used in order to find the success ratio of a specific chapter given the times a user failed and succeeded at it.

Student.java class contains a students (or not necessarily students but anyone's who wants to learn python operators) attributes, as well as getters and setters for them. A user has the following attributes: email, password, first_name and last_name. All of those attributes are stored in a MySQL database once the user registers.

Package database

In this package, we have created a single file called Dao.java which is a class consisting of all the methods used in order to either get data from the database or to store data in it. Dao.java class functions as the Controller of the MVC standard for our project. Each function is similar – at first the connection with the local database we have created is established by a connection string and after that by prepared statements we do whatever we need to do with the specific function regarding the database – either retrieve data from it, or push data in it.

Finally, in most functions we return a variable or an array which consists of the data we needed in order to display it in our jsp pages.

Package servlets

In the servlets package we store the java servlets used in our project. Login.java is responsible for getting the users data once he tries to log in, validate his credentials by calling a Dao.java method and after that redirect him to the appropriate page, or in case his credentials are not correct, show him an error message.

Register.java is responsible for getting the users data once he fills in the registration form and after that, by calling a Dao.java method store those data in the database. Finally, the user is redirected to the login page so that he can log in using his newly created credentials.

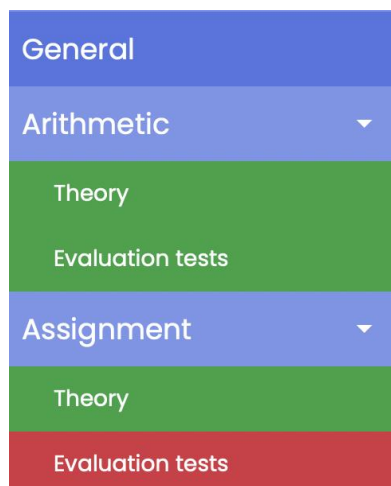
TestResult.java servlet is used so that we can calculate the results of a test submitted by the user and redirect him to a jsp page displaying those results. Based on which test option the user submitted, in the TestResult servlet initially by calling a Dao function, we store the result of the test in the database (whether the user failed or not) and after that the user is redirected to the corresponding jsp page displaying the results of the test he submitted.

Web Content

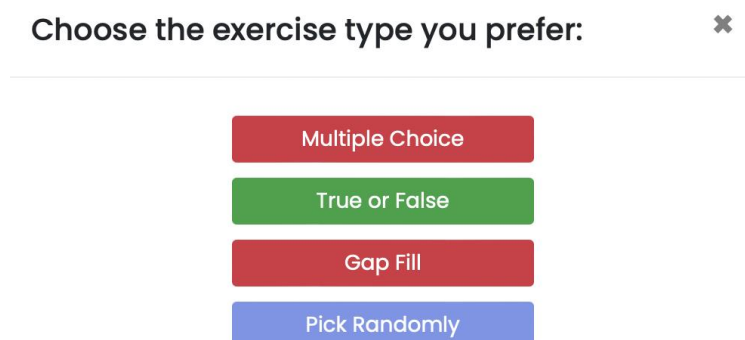
In the Web Content folder are stored all the views of the project, the JavaScript files used for various purposes as well as .css files used for styling and frontend. In the Javascript folder, there are three JavaScript files whose functionality is the following:

- confirmPassword.js: this file is used in the registration form in order to check whether the fields password and confirm_password the user entered are identical. If the two fields match, the registration process continues, otherwise an error message is displayed.
- loadSidebar.js: used in order to load the sidebar for every jsp page so that we don't have to manually add it in each page. In the onload function of the body tag of each jsp file in which we want to display the sidebar, we call the loadHtml() method of the loadSidebar.js file, which is responsible for displaying the sidebar which can be found on the SidebarTemplates/studentSidebar.jsp path.
- Sidebar.js: user for the toggle sidebar button. If the user presses the toggle sidebar button the sidebar collapses so that the user has more space for the other functions of the web project.

Continuing from the last file we mentioned above, the studentSidebar.jsp file contains the html-jsp code which represents the sidebar we have in our project. This file contains a part of java code for coloring purposes. Initially by calling the visitedOrNot() Dao function, we store in an array which chapters have been visited by the user and which have not been visited yet. If the chapter has been visited, its corresponding list element is colored in green, otherwise it's colored in red. Below we can see a part of the sidebar displaying what we explained earlier. In the following case, both Theory and all of the test chapters of Arithmetic Operators have been visited by the user, while on the other hand for Assignment Operators the user has not submitted each one of the test options.



Apart from the chapters, when a user clicks the evaluation tests option of each chapter, a modal appears (modals code is also implemented in this studentSidebar.jsp file) showing the test options which are also colored accordingly, depending on whether they have been submitted before or not. The test options modal can be seen in the below picture. In this case, the user has only submitted the True or False test options for that specific chapter.



The first file which gets loaded once the user logs in is the student.jsp. In this file, if the user has already submitted some tests before, a warning message is shown. In this message we warn the user about the chapter his success ratio is the worst compared to his other tests, and suggest him to read it's theory and then try the tests again in order to get better. This message can be seen below. In case the user has not taken any tests, of course this message does not appear at all.

Hello Giannis Gewrgiou!

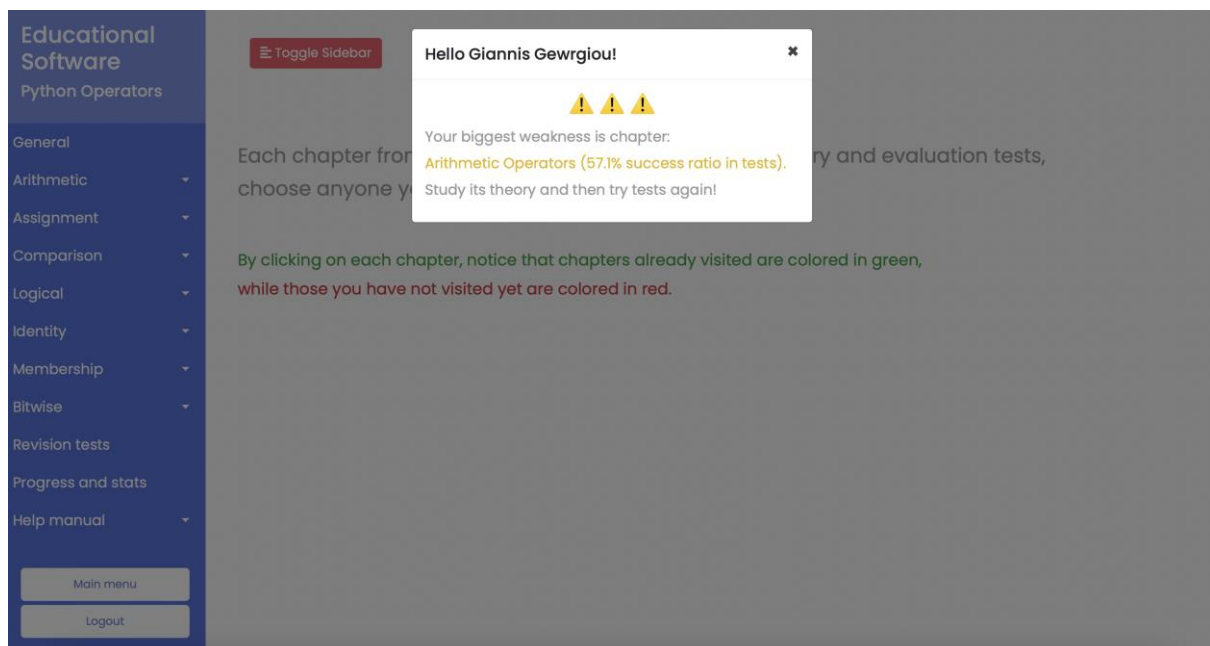


Your biggest weakness is chapter:

Arithmetic Operators (57.1% success ratio in tests).

Study its theory and then try tests again!

After the user closes this warning message, a “start” page is displayed which consists of the applications sidebar, along with some basic information about the applications functionality on the right side. The initial page can be seen below for a student that has taken tests and for a student that has not submitted any tests.



Educational Software
Python Operators

General
Arithmetic
Assignment
Comparison
Logical
Identity
Membership
Bitwise
Revision tests
Progress and stats
Help manual

Main menu
Logout

Toggle Sidebar

Hello Mixelis Andreadis!

Each chapter from the sidebar on the left consists of theory and evaluation tests, choose anyone you want to start learning!

By clicking on each chapter, notice that chapters already visited are colored in green, while those you have not visited yet are colored in red.

Continuing with the project views files, every chapter consists as we said of theory and evaluation tests. The chapter theory for each one is stored in a file called `chapte1_theory.jsp`, `chapter2_theory.jsp`, ..., `chapter7_theory.jsp`. All those files contain an explanatory table along with a paragraph and are responsible for displaying the theory part of the project to the user. Each time a file of those is visited, the database is updated regarding the times a specific chapter has been visited. Below some sample theory chapters are shown.

Educational Software
Python Operators

General
Arithmetic
Assignment
Comparison
Logical
Identity
Membership
Bitwise
Revision tests
Progress and stats
Help manual

Main menu
Logout

Toggle Sidebar

Arithmetic operators

are used with numeric values to perform common mathematical operations:

Operator	Name	Example
+	Addition	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/	Division	x / y
%	Modulus	$x \% y$
**	Exponentiation	$x ** y$
//	Floor division	$x // y$

Educational Software
Python Operators

General
Arithmetic
Assignment
Comparison
Logical
Identity
Membership
Bitwise
Revision tests
Progress and stats
Help manual

Main menu
Logout

Toggle Sidebar

Assignment operators are used to assign values to variables:

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3
&=	x &= 3	x = x & 3
=	x = 3	x = x 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

Once the user chooses a test option, or once he presses the pick randomly button which can be seen in the modal picture, he is redirected based on his choice to a specific jsp page containing the test questions. For chapter 1 and chapter 2 that we have implemented the evaluation tests, file containing them are: chapter1_multipleChoice.jsp, chape1_trueOrFalse.jsp, chapter1_gapFill.jsp and so on for chapter 2. Once the user is redirected to one of those pages, using the Dao.pickRandomly() function we randomly pick 5 questions from the database for the specific chapter to display to the user. In the database, we have stored 10 questions for each test option and for each chapter and each time we randomly pick 5 of them. It is obvious that the more questions stored in the database, the better it would be, but just for our projects purposes storing 10 questions was enough.

After the user submits the test, he is redirected to either the testResult.jsp or to the testResult_trueOrFalse.jsp page by the TestResult.java servlet which we analyzed earlier, based on which test he submitted. Multiple choice and gap fill tests, redirect the user to the testResult.jsp page, while true or false tests will redirect the user to the testResult_trueOrFalse.jsp page. This happens because the results of the tests in either case are displayed in different ways so for this reason we had to create two separate files. In either of those files, the result of the test is displayed in colorful ways similar to the sidebar we explained before. If the users answer to the question was correct, it's colored in green, otherwise it's colored in red and in this case we also display either which would be the correct answer to the question or the description about why the users answer was wrong. In the top of those pages, a message is shown about whether the user succeeded or failed at the test. Below we can see some photos regarding the tests as well as the pages displaying the results.

Educational Software

Python Operators

General

Arithmetic

Assignment

Comparison

Logical

Identity

Membership

Bitwise

Revision tests

Progress and stats

Help manual

Main menu

Logout

Toggle Sidebar

Multiple choice test - Select the answer you think is correct from the options given.

Q.1) Which operator should we use to multiply values 5 and 10?

- ☐ +
☒ *
☐ %

Q.2) Using which operator is floor division accomplished in python?

- ☐ %%
☒ //
☐ ^

Q.3) Choose the correct command to print the result of the subtraction of 5 from 50:

- ☐ print(50-5)
☒ print(50/5)
☐ print(5--50)

Educational Software

Python Operators

General

Arithmetic

Assignment

Comparison

Logical

Identity

Membership

Bitwise

Revision tests

Progress and stats

Help manual

Main menu

Logout

Toggle Sidebar

True or false test - Select true or false whether you think the statement is correct or not.

Q.1) The command `x-=3` will have the same output as `x=x-3`?

- ☐ True
☒ False

Q.2) Writing `z=z-3` is the same as writing `z-=3`.

- ☒ True
☐ False

Q.3) The command `y-=5` will decrement y's value by 5.

- ☐ True
☒ False

Q.4) The command `x-=3` will have the same output as `x=x-3`?

Educational Software

Python Operators

General

Arithmetic

Assignment

Comparison

Logical

Identity

Membership

Bitwise

Revision tests

Progress and stats

Help manual

Main menu

Logout

Toggle Sidebar

True or false test – Select true or false whether you think the statement is correct or not.

Q.1) The command `x-=3` will have the same output as `x=x-3`?

☐ True

☒ False

Q.2) Writing `z=z-3` is the same as writing `z-=3`.

☒ True

☐ False

Q.3) The command `y-=5` will decrement y's value by 5.

☐ True

☒ False

Q.4) The command `x-=3` will have the same output as `x=x-3`?

Educational Software

Python Operators

General

Arithmetic

Assignment

Comparison

Logical

Identity

Membership

Bitwise

Revision tests

Progress and stats

Help manual

Main menu

Logout

Toggle Sidebar

Fill the gap test – Complete the sentence by filling the gap.

Q.1) has the same result as `x//=100`.

Q.2) Writing is the same as writing `z-=3`.

Q.3) The command will decrement y's value by 5.

Q.4) has the same output as `y=y*10`.

Q.5) The command will have the same output as `x=x+5`.

Educational Software

Python Operators

General

Arithmetic

Assignment

Comparison

Logical

Identity

Membership

Bitwise

Revision tests

Progress and stats

Help manual

Main menu

Logout

Toggle Sidebar

TEST FAILED! 😞

Correct answers are colored in green and wrong answers are colored in red.

Q.1) Which operator should we use to multiply values 5 and 10?

Q.2) Using which operator is floor division accomplished in python?

Q.3) Choose the correct command to print the result of the subtraction of 5 from 50:

Your answer: `print(50/5)`

Correct answer: `print(50-5)`

Q.4) Which operator should we use to multiply values 5 and 10?

Your answer: `%`

Correct answer: `*`

Educational Software

Python Operators

General

Arithmetic

Assignment

Comparison

Logical

Identity

Membership

Bitwise

Revision tests

Progress and stats

Help manual

Main menu

Logout

TEST SUCCEEDED! 😊

Correct answers are colored in green and wrong answers are colored in red.

Q.1) Writing `z=z+10` is the same as writing which of the following commands?

Your answer: `x*=100`

Correct answer: `z+=10`

Q.2) Choose the correct command to multiply x by 20.

Your answer: `x=x+20`

Correct answer: `x*=20`

Q.3) Choose the correct command to print the result of the subtraction of 5 from 50:

Q.4) Which command will have the same output as `x=x|5`?

Q.5) Choose the correct option to to decrement y's value by 5.

By using all of the questions stored for the chapters we explained earlier, we also create revision tests which the user can select. Revision tests option can be found in the sidebar along with all the other chapters. After the user submits a revision test, he is similarly redirected to the testResult.jsp page by the TestResult.java servlet, in which we display the results of his test.

The last mentionable file of the Web Content folder is the progressAndStats.jsp file. In this file we display all of the users stats in separate organized div elements, as well as some explanatory graphs about the users progress and stats in each chapter in the end of the page. When this file is called after the user presses the corresponding button on the sidebar, at first it retrieves all the stats of the specific user from the database by calling the Dao.retrieveTheoryStats() and Dao.retrieveTestStats() functions and after that it calculates the success ratio for each chapter which is also displayed. For the chapters 3-7 we only display how many times each theory chapter has been visited as we have not implemented test chapters for them. Implementing the test chapters just for our two first chapters would be enough for the purposes of our project and implementing tests for all of the chapters would be a really easy to do but extremely time consuming process as for each chapter we have to add totally 30 questions manually in the database. Below some photos for a user we have created for testing purposes can be seen. It should be noted that next to each chapters success ratio, either a happy or a sad emoji is displayed. If the users ratio is higher than 50% a happy emoji is displayed, otherwise a sad one.

Educational Software
Python Operators

General

Arithmetic

Assignment

Comparison

Logical

Identity

Membership

Bitwise

Revision tests

Progress and stats

Help manual

Main menu

Logout

Choose any chapter from the list below to jump straight into its stats, or just scroll.

Chapter 1 - Arithmetic Operators

Chapter 2 - Assignment Operators

Chapters 3 - 7 Theory Stats

Revision Tests

Explanatory Graphs

CHAPTER 1 - ARITHMETIC OPERATORS

Multiple Choice

Times succeeded: 15

Times failed: 13

Success ratio: 53.6% 😊

True or False

Times succeeded: 3

Times failed: 4

Success ratio: 42.9% 😞

Gap Fill

Times succeeded: 3

Times failed: 1

Success ratio: 75.0% 😊

Theory

Times visited: 75

Educational Software

Python Operators

General

Arithmetic

Assignment

Comparison

Logical

Identity

Membership

Bitwise

Revision tests

Progress and stats

Help manual

Main menu

Logout

CHAPTER 2 - ASSIGNMENT OPERATORS

Multiple Choice

Times succeeded: 1

Times failed: 0

Success ratio: 100.0% 😊

True or False

Times succeeded: 3

Times failed: 3

Success ratio: 50.0% 😞

Gap Fill

Times succeeded: 1

Times failed: 0

Success ratio: 100.0% 😊

Theory

Times visited: 23

THEORY CHAPTERS - TIMES VISITED EACH

CHAPTER 3 - COMPARISON OPERATORS: 11

CHAPTER 4 - LOGICAL OPERATORS: 11

CHAPTER 5 - IDENTITY OPERATORS: 9

CHAPTER 6 - MEMBERSHIP OPERATORS: 12

CHAPTER 7 - BITWISE OPERATORS: 12

General

Arithmetic

Assignment

Comparison

Logical

Identity

Membership

Bitwise

Revision tests

Progress and stats

Help manual

Main menu

Logout

REVISION TESTS

Times succeeded: 2

Times failed: 2

Success ratio: 50.0% 😞

Total Percentage of Tests Success Ratio per Chapter

Times Visited Theory Chapters

General

Arithmetic

Assignment

Comparison

Logical

Identity

Membership

Bitwise

Revision tests

Progress and stats

Help manual

Main menu

Logout

Success Ratio of Chapter 1 Different Test Types

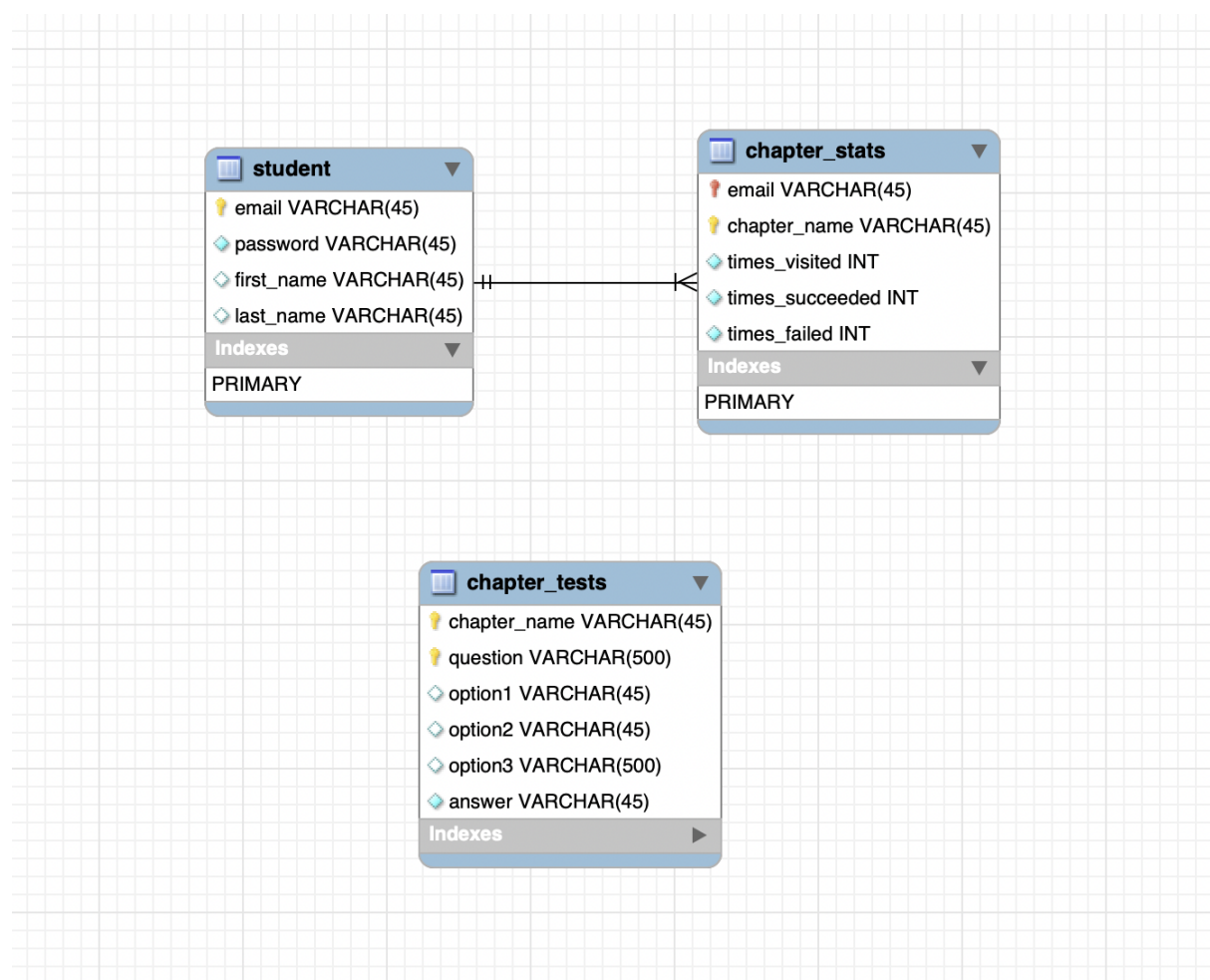
Success Ratio of Chapter 2 Different Test Types

Revision Tests

For our database, we created a schema called educationalsoftware which consists of the three following tables, which will later be explained:

- student (email, password, first_name, last_name),
- chapter_stats (email, chapter_name, times_visited, times_succeeded, times_failed),
- chapter_tests (chapter_name, question, option1, option2, option3, answer).

The ER diagram of the database we created, using MySQLWorkbench, can be seen in the picture below:



The table “student” consists of the following columns: email, password, first_name and last_name. All those data are stored in the database after the users registration process is completed. As we already mentioned, students password is saved in the database after it’s encrypted by using the MD5 hashing algorithm.

The table chapter_stats consists of the following columns: email, chapter_name, times_visited, times_succeeded and times_failed. The email column stands for the unique users email, chapter_name stands for a specific chapter (e.g. chapter1_theory, chapter2_multipleChoice etc.), times_visited stands for how many times the specific user has

visited the chapter, while `times_succeeded` and `times_failed` stand for how many times the user has succeeded or failed in the specific test, respectively. The two last columns, `times_succeeded` and `times_failed` are used only for evaluation test chapters and not for theory chapters as theory cannot be failed or succeeded. For theory chapters, those two columns are given the value 0.

The table `chapter_tests` consists of the columns `chapter_name`, `question`, `option1`, `option2`, `option3` and `answer`. This table is used for storing evaluation test questions, options for each question so that the multiple choice or true or false tests can be created, as well as the correct answer for each question. For all kind of evaluation test options (multiple choice, true or false, gap fill), the question is stored in the question column and the correct answer is stored in the answer column. For multiple choice tests, `option1`, `option2` and `option3` columns represent the three possible options that are given to the user to choose from. For true or false tests, `option1` is always "True", `option2` is always "False" and in `option3` column we store the description of what would be correct for that question, in order to display it in case the user does not respond correctly. Lastly, in gap fill evaluation tests, `option1`, `option2` and `option3` columns are empty as there are no possible options to display in fill-the-gap tests and we only need columns `chapter_name`, `question` and `answer`.