Faculty of Organizational Sciences

Software Quality & Testing Laboratory

dr. Robert Leskovar

Computer science and informatics

Vuk Stojkovic

September 26, 2021

Statement
Student Vuk Stojković of the UN study program Organization and Management of Information Systems declares: I am the author of the solutions to all tasks assigned to me within the subject Computer Science and Informatics. I guarantee that:

- I have written the solutions myself and that I have not used the solutions of other authors without appropriate references;
- I have made sure that the works and opinions of other authors I use in this document are cited and cited;
- I have made sure that the works and opinions of other authors listed in the list of sources are written in accordance with the faculty instructions;
- I am aware that plagiarism is the presentation of other works, either in the form of a quotation or in the form of an almost literal paraphrasing, or in a graphic or tabular form, with which foreign thoughts or ideas presented as my own - punishable by law (Copyright and Related Rights Act), followed by measures taken by the Faculty of Organizational Sciences of the University of Maribor in accordance with its rules;
- I am aware of the consequences that proven plagiarism can have for the submitted document and for my status at the Faculty of Organizational Sciences of the University of Maribor.

As a participant in the course Computer Science and Informatics, I will follow (a) general ethical and professional norms:

- contribute to the well-being of society and humanity;
- avoided harming others;
- I will be honest, trustworthy and take action against discrimination;
- respect intellectual property rights including copyrights and licenses;
- respect intellectual property rights even when they are not specifically protected;
- respect the privacy of others;
- consistently implement the principle of confidentiality, except where the law obliges me to disclose;
- strive for the highest quality, efficiency and dignity both in the manufacturing process and in my products;
- to the best of my ability to acquire and maintain professional competence;
- to respect contracts, agreements and assigned responsibilities - to use information and communication technology only when I am authorized to use;
- considered the violation of the above norms as unacceptable and incompatible with the status of a participant in the subject Computer Science and Informatics.

Contents

1 Theoretical explanations

1.1 Computer architecture & performances

1.2 GPS operation

1.3 Big Data

1.4 XML creation and syntax test

2 Development of OracleDB application

Estimated time consumption in hours: 3 hours
Actual time consumption in hours: 3 hours

- Create a free workspace for Oracle Apex
- In Apex, create your own table, the columns of which are related to some aspect of the company's business (eg data on employees, machines, operations performed by machines, fixed assets, salaries, orders, price lists, etc)
- Create a desktop application. Add only a report and a Report and Form to the application that refers to the table from the previous point. Enter some test data (at least 5 records).

SQL script for creating table

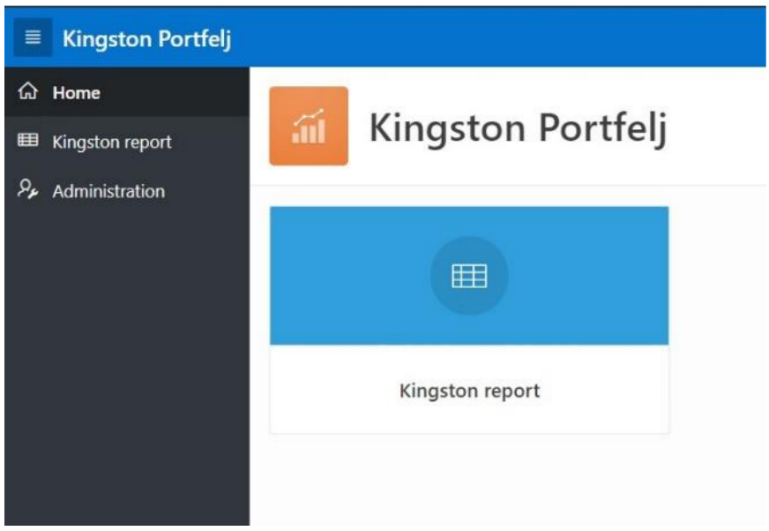

Figure 2: Database Application GUI

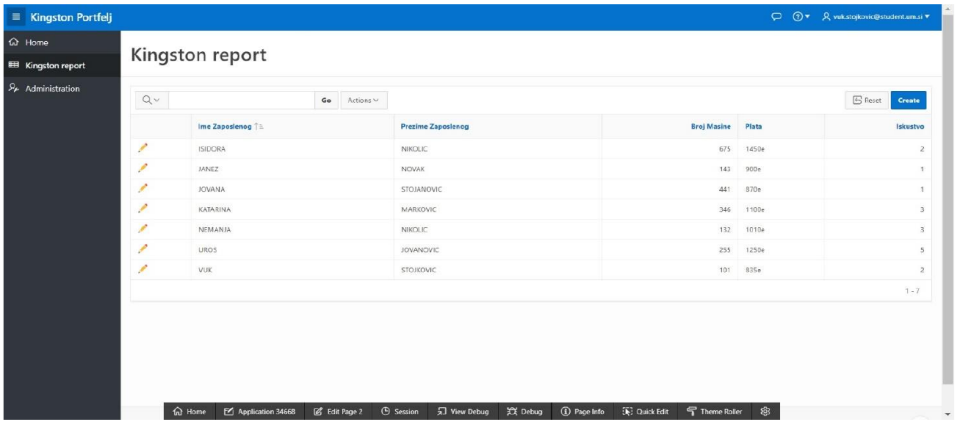

Figure 3: Database Application report

Figure 4: Database Application adding new user

3 PGP Encryption

Estimated time consumption in hours: 0.5 hour
Actual time consumption in hours: 0.5 hour

Firstly I'll set up the GnuPrivacy Guard software on Kubuntu 20.04 LTS. After that, I'll generate and export pair of keys for my server, and repeat those two steps on virtual machine Windows 10, using Cleopatra software.

```
$ sudo apt-get install haveged
$ gpg --version
$ gpg --gen-key
$ gpg --list-key
```

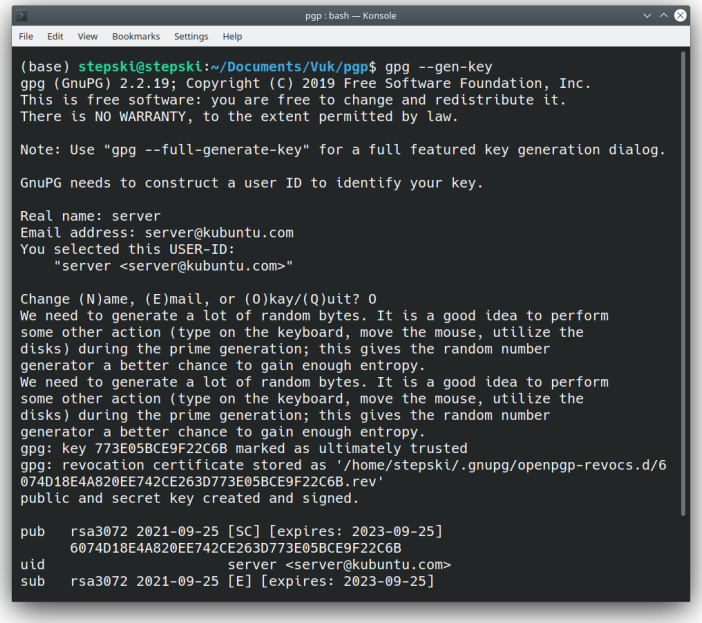

Figure 5: Generating keys at server

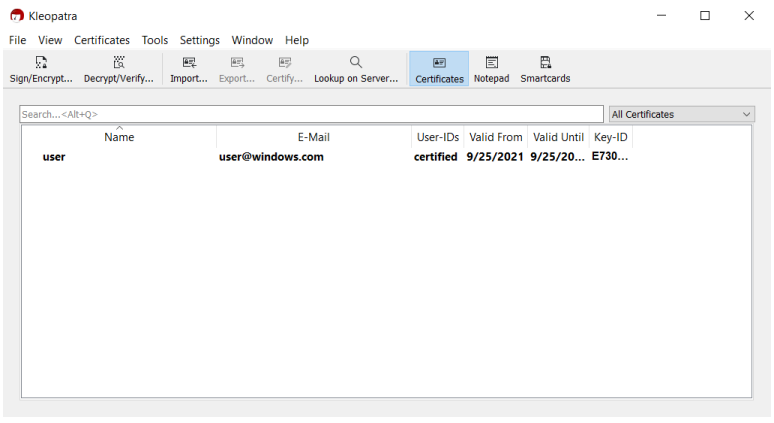

Figure 6: Generating keys for user

Next step is to import the user public key on Linux server, and use the credentials for encrypting the wanted file - in this case a simple pdf.

```
$ gpg --armor --export server@kubuntu.com > serverPublicKey.asc
$ gpg --import userPublicKey.asc
$ gpg -e -r user@windows.com --output lockedPdf.gpg --encrypt Jackknife_analysis.pdf
```

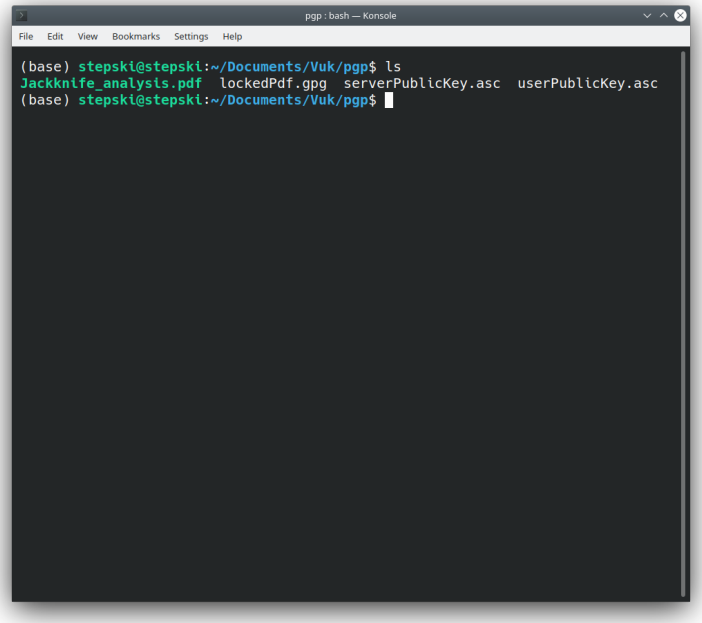

Figure 7: List of files in pgp folder on server

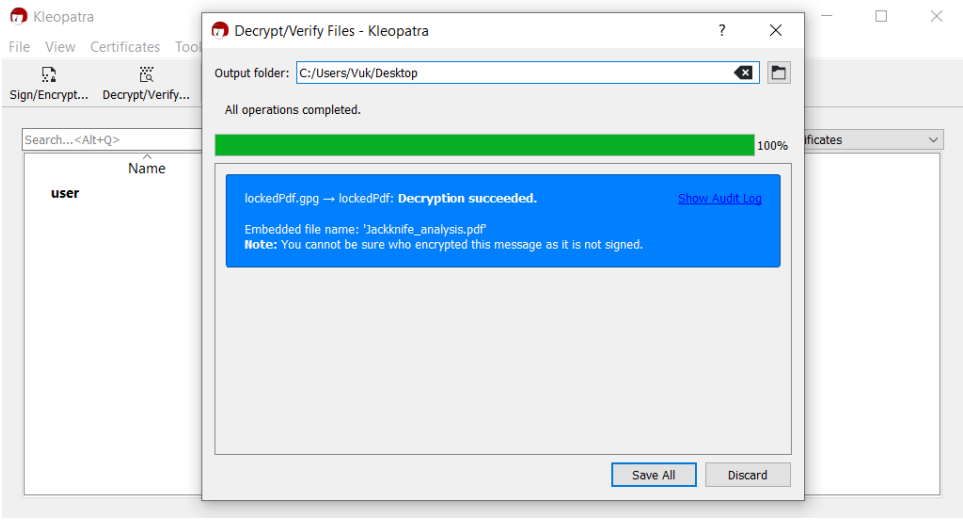

Figure 8: Decrypting file via Cleopatra

4 Microprocessor Simulator SMS32v50

Estimated time consumption in hours: 3 hour
Actual time consumption in hours: 2 hour

The task is to write an program in simulator SMSv32. The program should calculate the sum of numbers from 12 inclusive (decimal value) to 19 inclusive (decimal value) by using a loop. Use the registry of your choice to assign an initial value. The magnification step for loop control is 1. Break the loop when the condition is fullfiled. Create two versions of the program, using JNZ for one and JNS for the other.
Assembly code:

```
MOV AL, 13
MOV BL, 13
LOOP: DEC AL
ADD BL, AL
CMP AL, D
JNS LOOP
END
```

```
MOV AL, C
MOV BL, C
LOOP: INC AL
ADD BL, AL
CMP AL, 13
JNZ LOOP
END
```

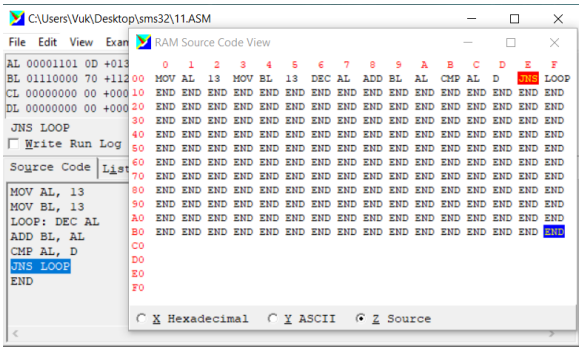

Figure 9: JNS 17B

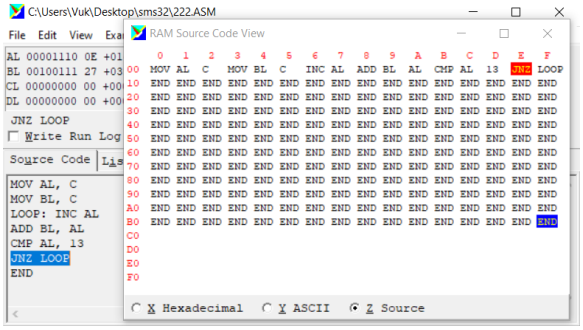

Figure 10: JNZ 17B

5 MySQL Database

Estimated time consumption in hours: 4 hour
Actual time consumption in hours: 2 hour

Create your data schema (mysql calls schemas "databases") and 4 tables in the mysql database management system in the virtual machine. The schema named "tantalum" should support the editing specified by utf8-general_ci, and the tables the utf8 code set. You can use any interface (e.g. terminal window, phpmyadmin, mysql Workbench, etc.) to manage the database.
Table 1: the data refer to business areas in the company. Attributes are:

- sifra_poslovnega_podrocja; varchar(2),
- naziv_poslovnega_podrocja; varchar(100).

The data is comma separated and is in the same order as the following attributes:

- A: Administracija
- F: Finance in računovodstvo
- M: Marketing
- N: Nabava
- P: Proizvodnja

Table 2: data refer to the organizational structure in a company. Use the script below to create the table and data.

SQL script for creating tables

Table 3: Data refer to business processes. Use the script below to create the table and data.

SQL script for creating tables and inserting data

Table 4: Data refer to business process levels. Use the script below to create the table and data:

SQL script for creating tables and inserting data

Write 5 queries to add to the PROCES_NOVI table. Process IDs should be between 1000 and 1004, the level can only be in the set of already existing ones. You can invent the name of the process, the local weight should be 0, and the business area should be in the set of already existing ones. Run all 5 queris at once and represent the result.

```
INSERT INTO PROCES_NOVI VALUES (1000, 'IZV', 'primer procesa izvedbenega nivoja', 0, 'A');
INSERT INTO PROCES_NOVI VALUES (1001, 'KON', 'primer procesa kontrolnega nivoja', 0, 'F');
INSERT INTO PROCES_NOVI VALUES (1002, 'OPP', 'primer procesa na nivoju operativnega planiranja',
0, 'M');
INSERT INTO PROCES_NOVI VALUES (1003, 'TAK', 'primer procesa na takticnem nivoju', 0, 'N');
INSERT INTO PROCES_NOVI VALUES (1004, 'STR', 'primer procesa na strateskem nivoju', 0, 'P');
```

+ Options

sifra	nivo	naziv	lokutez	poslovno_podrocje
503	IZV	Pregled planiranih proizvodnih nalogov	3	P
504	IZV	Pregled planiranih skladiščnih nalogov	3	P
505	IZV	Pregled poročil o zalogah	3	N
506	IZV	Pregled kadrovskih poročil	3	A
1000	IZV	primer procesa izvedbenega nivoja	0	A
1001	KON	primer procesa kontrolnega nivoja	0	F
1002	OPP	primer procesa na nivoju operativnega planiranja	0	M
1003	TAK	primer procesa na takticnem nivoju	0	N
1004	STR	primer procesa na strateskem nivoju	0	P

Figure 14: Output

Next task is to write queries for changing records in the PROCES_NOVI table:

- process with IDs between 1000 and 1004, change the local weight to 100
- process with ID 1002 change the name to "Zombie process"
- process with ID 1004 change the business area to "A". Present each command and the result of the execution in a report.

```
UPDATE PROCES_NOVI SET lokutez=100 WHERE sifra BETWEEN 1000 and 1004;
UPDATE PROCES_NOVI SET naziv = 'Zombi proces' WHERE sifra = 1002;
UPDATE PROCES_NOVI SET poslovno_podrocje = 'A' WHERE sifra = 1004;
```

+ Options

sifra	nivo	naziv	lokutez	poslovno_podrocje
503	IZV	Pregled planiranih proizvodnih nalogov	3	P
504	IZV	Pregled planiranih skladiščnih nalogov	3	P
505	IZV	Pregled poročil o zalogah	3	N
506	IZV	Pregled kadrovskih poročil	3	A
1000	IZV	primer procesa izvedbenega nivoja	100	A
1001	KON	primer procesa kontrolnega nivoja	100	F
1002	OPP	Zombi proces	100	M
1003	TAK	primer procesa na takticnem nivoju	100	N
1004	STR	primer procesa na strateskem nivoju	100	A

Figure 15: Output

Now write the queries for removing the rows in table PROCES_NOVI:

- remove process with ID 1001
- remove process with ID 1000 or 1002
- remove process with ID greater than 999

```
DELETE FROM PROCES_NOVI where sifra=1001;
DELETE FROM PROCES_NOVI where sifra=1000;
DELETE FROM PROCES_NOVI where sifra=1002;
DELETE FROM PROCES_NOVI WHERE sifra>999;
```

+ Options

sifra	nivo	naziv	lokutez	poslovno_podrocje
503	IZV	Pregled planiranih proizvodnih nalogov	3	P
504	IZV	Pregled planiranih skladiščnih nalogov	3	P
505	IZV	Pregled poročil o zalogah	3	N
506	IZV	Pregled kadrovskih poročil	3	A

Figure 16: Output

In next few queries list following:

- all attributes of all processes
- all attributes of all processes that have a ID between 100 and 150
- all attributes of all processes that have a ID between 100 and 150. Arrange the list in alphabetical order of names.
- all attributes of all processes that have a level equal to “STR” and business area “P”. Arrange the print in alphabetical order of business areas and titles.
- all attributes of all processes that have a business scope between “A” and “F”. Arrange the print in alphabetical order of business areas and titles.
- the level and number of processes of each level
- the business area and the number of processes of each business area
- the sum of all local weights
- the minimum, maximum and average value of local weights
- the sum of local weights by business areas

```
select * from PROCES_NOVI;
select * from PROCES_NOVI where sifra between 100 and 150;
select * from PROCES_NOVI where sifra between 100 and 150 order by naziv;
select * from PROCES_NOVI where nivo ='STR' and poslovno_podrocje = 'P'
order by poslovno_podrocje, naziv;
select * from PROCES_NOVI where poslovno_podrocje between 'A' and 'F'
order by poslovno_podrocje, naziv;
select nivo, count(nivo) from PROCES_NOVI group by nivo;
select poslovno_podrocje, count(poslovno_podrocje) from PROCES_NOVI group by poslovno_podrocje;
select sum(lokutez) from PROCES_NOVI;
select min(lokutez), max(lokutez), avg(lokutez) from PROCES_NOVI;
select poslovno_podrocje, sum(lokutez) from PROCES_NOVI group by poslovno_podrocje;
select nivo, sum(lokutez) from PROCES_NOVI group by nivo;
```

6 C++ programming

Write the following program: The default constructor function enters the sum of the row index increased by 2 and the column index increased by 3 into the matrix element "v". The default constructor function, which has a parameter, should enter the sum of the row index increased by the parameter value "A" and the index of the column increased by the value of the parameter "a". The function "izpis_vse" before the value of the element of the matrix "v" printed the index of the row and column, for example: (row, column) the value of the element and the space. The main function declared object "y" with parameter 17. On object y use the function "izpis_vse" - in the main function. The main function (main) declared a vector of objects "z" of length 10 elements. All vector objects should be created with a default constructor without a parameter. Use the "izpis_vse" function on the fourth element of the "z" object.

```
#include <iostream>
using namespace std;
    class tbl {
        public: int v[3][3];
        tbl () {
            for(int i=0; i < 3; i++)
            for(int j=0; j < 3; j++) v[i][j]= (i+2)*(j+3);};
            tbl (int a) {
                for(int i=0; i < 3; i++)
                for(int j=0; j < 3; j++) v[i][j]= (i+a)+(j+a);};
                void izpis_vse (){
                    int i=0; int j=0;
                    cout<< " \n";
                    while (i < 3) {
                        while (j < 3)
                        {cout<<"("<<i<<","<<j<<")"<<v[i][j]<<" "; j++;}
                        j=0; i++;}};};

                        int main()
                        {
                            tbl x, y(17), z[10];
                            x.izpis_vse();
                            y.izpis_vse();
                            z[3].izpis_vse();
                            return 0;

                        }
```

7 Java programming

The function "generiraj_nakljucno" had an integer parameter that will determine the number of iterations of the for loop in this function. Use the provided code and correct the call of this function in the main function (main) and display on the screen - write how many random numbers were generated (instead of "million" insert the calculation). Correct the for loop in the main program so that the number of iterations determines the integer parameter to which you assign a value before executing the loop. Correct the display on the screen - write down how many random numbers were generated (instead of 1000 * insert the correct calculation formula). Print the function named "izpis_rezultata". This function is private static void. It should have three parameters: the number of for loop iterations in the main program, the number of for loop iterations in the "generiraj_nakljucno" function, and the result of the calculation ((100 * (dinamicni_pi / Math.PI)) - 100). The function should display:

```
Pi je približno xx pri xx naključnih številih.
Odstopanje ocenjenega Pi od pravega je xx odstotkov.
Hvala za rože
```

Java code for randomPI

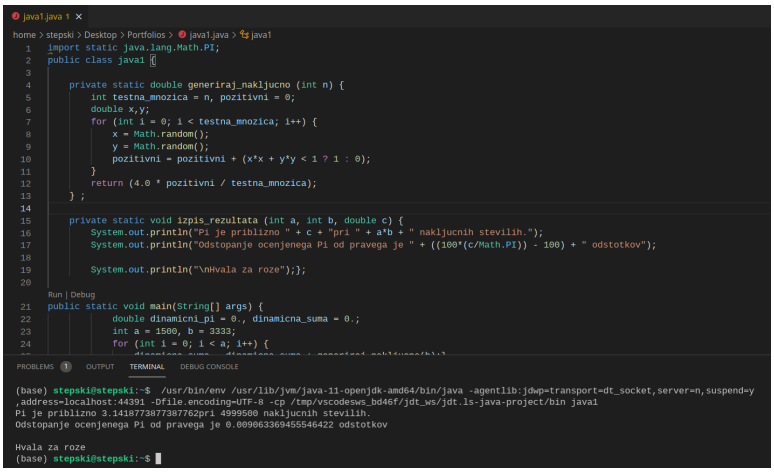

Figure 18: Output

For the second program use the providen code and change the program in Java so that it accepts only the following values: number of participants greater than 1, starting position between 0 and number of participants minus 1 and counting step greater than 1. The program must not continue until the user enters the appropriate values of variables Tthe program wrote the state of the count (vector circle) in a file named "potek.log".

Java code for counter

8 Web development using HTML, PHP & MYSQL

Based on the presented example, you will be able to develop your own web solution that will enable printing a list of all records in the table, viewing an individual record in the table, dding a new record to the table and, changing an existing one. The structure of the solution is shown in the diagram below:

Firstly create your own table in the virtual machine with the phpmyadmin tool. The table should have a defined key and a total of 5 attributes. Use the RII2016 schema in the database management system. Example of creating a table named files:

Using NetBeans IDE create a new project in PHP and choose any name for your project. In the Run As box, select PHP Built-in Web Server. The tool will prepare an index.php file.

Program other php files:

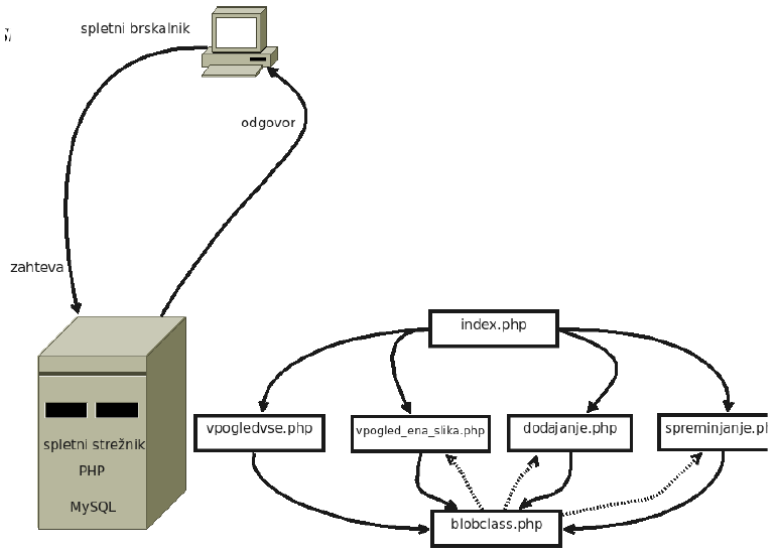

Figure 20: Output

SQL script for creating tables for web app

PHP script for creating application landing page

- vpogledvse.php : include the blobclass.php file. Declare the BobDemo class, and the $ blobObj object of this class. Call the function $ blobObj → selectall ().
- vpogled_ena_slika.php : include the blobclass.php file. Declare the BobDemo class, and the $ blobObj object of this class. Download the value of the code from the web form and enter it in the variable $ code. This is followed by a call to the member function $ blobObj → selectBlob ($ code) and the result is displayed on the screen. The result is stored in the variable $ a, which is a field (array) with all the attributes of one record in the database - also an image in the data type blob (the limit is 2 MB).
- dodajanje.php : include the blobclass.php file. The latter declare the BobDemo class, and the $ blobObj object of this class. If no file is selected (image in jpg. Png format, etc.), print the "Error:" and the error number will be displayed. If a file is selected, this file is temporarily stored in the / images subdirectory of the current directory. The file size information is displayed, followed by a call to the $ blobObj → insertBlob membership function, deletion of the temporary file and information that the addition was successful.
- spreminjanje.php : include the blobclass.php file. Declare the BobDemo class, and in this program declare the $ blobObj object of this class. If no file is selected (image in jpg. Png format, etc.), print the warning "Error:" and the error number will be displayed. If a file is selected, this file is temporarily stored in the / images subdirectory of the current directory. Prepare variables that will allow you to change the image and call the membership function $ blobObj → updateBlob. This is followed by deleting the temporary file and the information that the change was successful.

- bobclass.php : prepares commands for database manipulation (SELECT, INSERT, UPDATE). First, the constructor function __construct is defined, followed by the functions insertBlob, updateBlob, selectBlob and selectall, which are used by the programs vpogledvse.php, vpogled_ena_slika.php, dodajanje.php and spreminjanje.php. The object constructor will provide a connection to the database, using the RII2016 schema, the student username, and the StudentFov2016 + password. Other functions prepare SQL commands (prepare - with the data we entered in the web form) and execute this command.

PHP script for creating "BobDemo"

Create new folder and name it "images". Check if the directories structure looks as this:

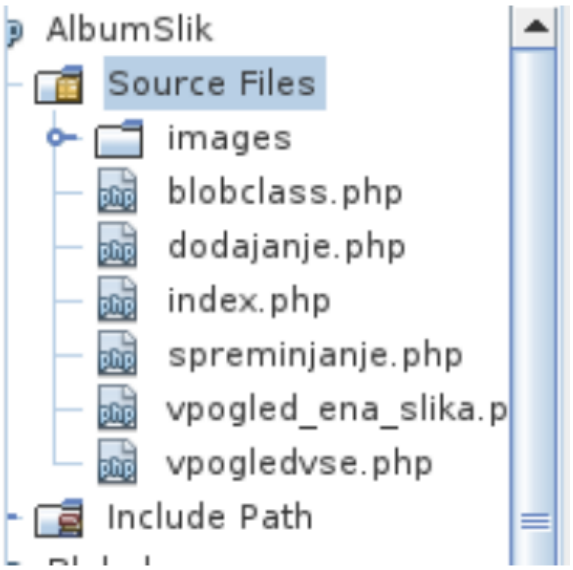

Figure 24: Project structure

PHP script for creating function "vpogledvse"

PHP script for creating function "vpogled_ena_slika"

PHP script for creating function ”dodajanje”

PHP script for creating function ”spreminjanje”

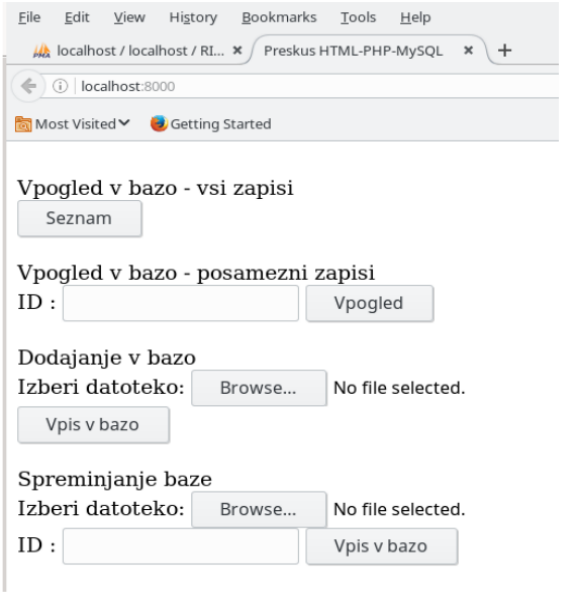

Figure 29: Web Application UI

If you click the list button, only the attributes of the table are displayed. However, no content has been added yet. Therefore, when adding, click the Browse button and select the image to upload to the database. After selecting the file, click the Register button. Upload at least 7 different images. Make sure you have successfully uploaded all the images you want by selecting Database View - Individual Records. Then replace one of the uploaded images with another - changing the database.

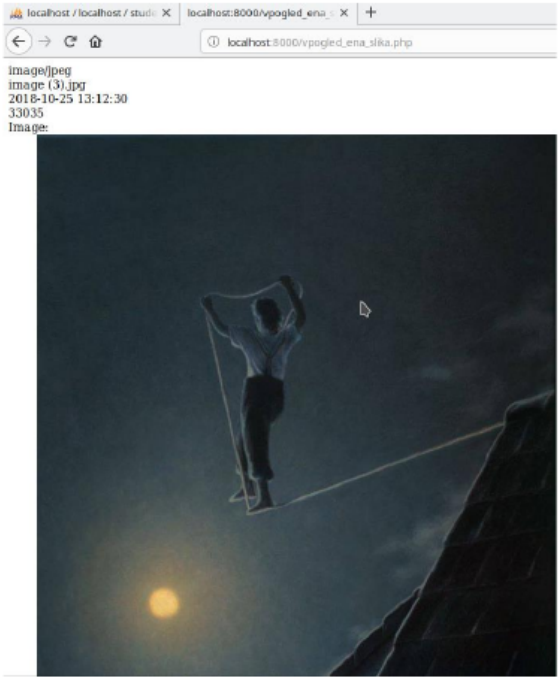

Figure 30: Web Application UI

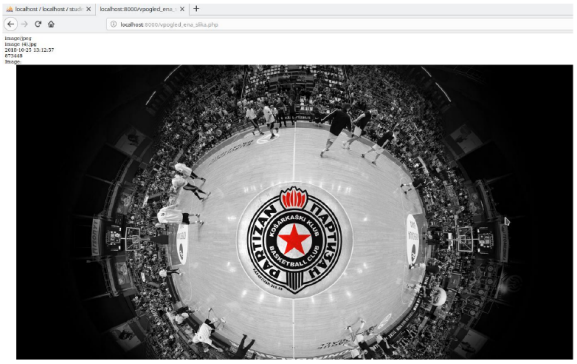

Figure 31: Adding photos to web app

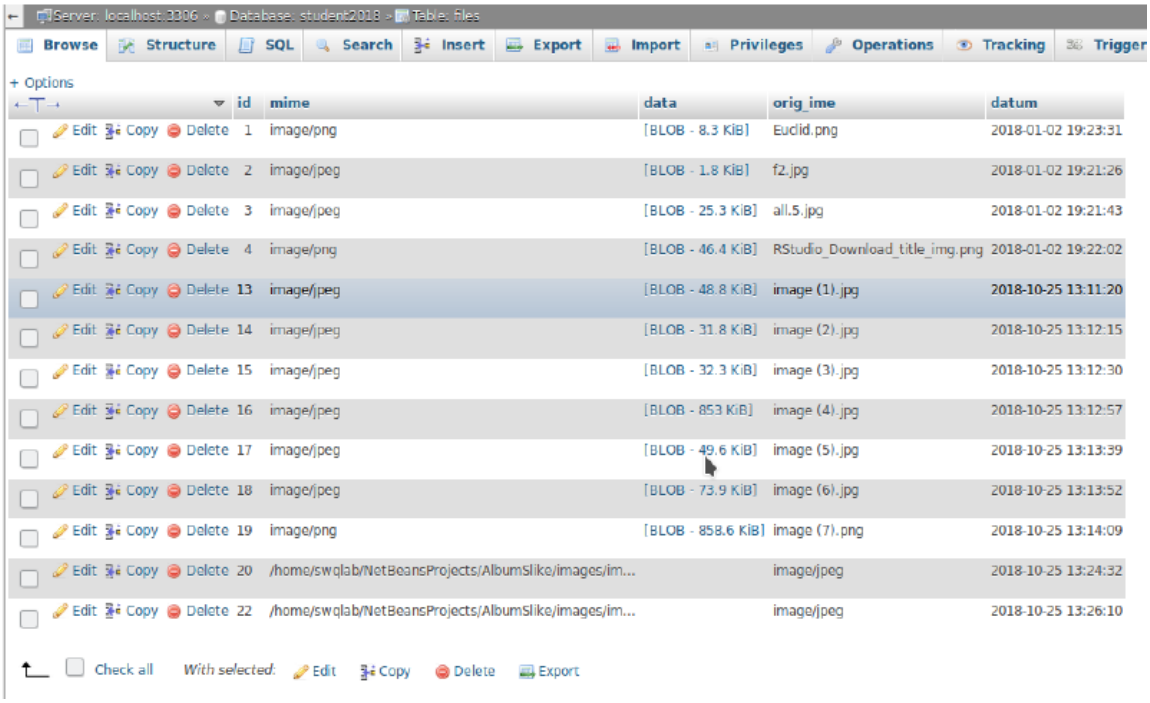

Figure 32: Web Application database system