MySQL Project:

Tomer Flom - 303015671

Adiel Perez - 308101062

Contents

[Tables: 2](#_Toc503557319)

[Application GUI screenshots: 3](#_Toc503557320)

[InitializeDB: 7](#_Toc503557321)

[Project Backend Source Code: 13](#_Toc503557322)

# Tables:

|  |  |  |  |
| --- | --- | --- | --- |
| **topic\_id** |  | Name | Speciallity |
| 1 |  | C++ | Debugger |
| 2 |  | C++ | Developer |
| 3 |  | JAVA | Frontend |
| 4 |  | JAVA | Backend |
|  | Etc… | | |

**software\_topics**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| engineer\_id | first\_name | last\_name | birthdate | address | *topic\_id* |
| 1 | Moshe | Cohen | 1989/5/17 | Blah… | 2 |
| 2 | … | .. | .. | .. | 3 |
| 3 | .. | .. | .. | .. | 2 |
| .. | .. | .. | .. | .. | .. |

**engineers**

|  |  |  |  |
| --- | --- | --- | --- |
| project\_id | projectName | clientName | startDate |
| 1 | NewBridge | John doe | 2015/1/13 |
| 2 | … | .. | .. |
| 3 | .. | .. | .. |
| .. | .. | .. | .. |

**projects**

|  |  |
| --- | --- |
| dev\_topic\_id | name |
| 1 | Configuration Management |
| 2 | Design |
| 3 | Requirements management |
| 4 | Encoding |
| .. | .. |

**development\_topics**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| milestone\_id | description | targetDate | amount | *project\_id* |
| 1 | M1 | 2016/2/2 | 0 | 1 |
| 2 | M2 | 2016/2/10 | 1500 | 1 |
| 3 | M3 | 2016/3/1 | 5800 | 1 |
| .. | .. | .. | .. |  |

**milestones**

|  |  |
| --- | --- |
| engineer\_id | phone |
| 1 | 052-3333355 |
| 1 | 08-8866212 |
| 2 | 053-9572223 |
| .. | .. |

**engineers\_phones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *engineer\_id* | *project\_id* | grade | month | Year |
| 1 | 1 | 10 | 1 | 2017 |
| 2 | 1 | 9 | 1 | 2017 |
| 1 | 2 | 5 | 1 | 2017 |
| 2 | 2 | 3 | 1 | 2017 |
| 1 | 1 | 9 | 2 | 2017 |
| .. | .. | .. | .. | .. |

**Grades**

|  |  |
| --- | --- |
| *engineer\_id* | *project\_id* |
| 1 | 1 |
| 1 | 2 |
| 2 | 1 |
| 2 | 2 |
| .. | .. |

**project\_participations**

|  |  |  |  |
| --- | --- | --- | --- |
| auto\_id | *project\_id* | *dev\_topic\_id* | tool |
| 1 | 1 | 1 | Github |
| 2 | 1 | 2 | amazon crm |
| .. | .. | .. | .. |

**development\_tools**

# Application GUI screenshots:

**A screenshot of a computer

Description generated with very high confidenceA screenshot of a computer

Description generated with very high confidenceA screenshot of a computer

Description generated with very high confidenceA screenshot of a social media post

Description generated with very high confidenceA screenshot of a cell phone

Description generated with very high confidenceA screenshot of a social media post

Description generated with very high confidenceA screenshot of a social media post

Description generated with very high confidenceA screenshot of a social media post

Description generated with very high confidence**

# InitializeDB:

<?php

$servername = "localhost";

$username = "mysql\_montv";

$password = "dinoflom";

// Create connection

$conn = new mysqli($servername, $username, $password);

// Create database

createDB($conn);

createTables($conn);

initializeValues($conn);

$conn->close();

?>

<?php

**function createDB($conn)**{

$sql = "CREATE DATABASE projectDB";

if ($conn->query($sql) === TRUE) {

echo "Database created successfully" . "<BR><BR>";

} else {

echo "Database already exists. please delete it and then try again. <BR>";

echo "or maybe try to procceed into the <a href='../GUI'>GUI</a>";

exit();

}

}

**function createTables($conn)**{

$conn->query("USE projectDB");

$newTable\_software\_topics = "CREATE TABLE software\_topics (

topic\_id INT UNSIGNED AUTO\_INCREMENT,

name VARCHAR(50) NOT NULL,

specialty VARCHAR(50) NOT NULL,

PRIMARY KEY (topic\_id)

) ENGINE=InnoDB";

$newTable\_engineers = "CREATE TABLE engineers (

engineer\_id INT UNSIGNED AUTO\_INCREMENT,

software\_topic\_id INT UNSIGNED NOT NULL,

firstname VARCHAR(50) NOT NULL,

lastname VARCHAR(50) NOT NULL,

birthdate DATE,

address MEDIUMTEXT,

PRIMARY KEY (engineer\_id),

FOREIGN KEY (software\_topic\_id) REFERENCES software\_topics (topic\_id)

ON DELETE RESTRICT

ON UPDATE CASCADE

) ENGINE=InnoDB";

$newTable\_projects = "CREATE TABLE projects (

project\_id INT UNSIGNED AUTO\_INCREMENT,

projectName VARCHAR(30) NOT NULL,

clientName VARCHAR(30) NOT NULL,

startDate DATE,

PRIMARY KEY (project\_id)

)ENGINE=InnoDB";

$newTable\_milestones = "CREATE TABLE milestones (

milestone\_id INT UNSIGNED AUTO\_INCREMENT,

project\_id INT UNSIGNED NOT NULL,

amount DOUBLE,

targetDate DATE,

description MEDIUMTEXT,

PRIMARY KEY (milestone\_id),

FOREIGN KEY (project\_id) REFERENCES projects (project\_id)

ON DELETE CASCADE

ON UPDATE CASCADE

) ENGINE=InnoDB";

$newTable\_development\_topics = "CREATE TABLE development\_topics(

dev\_topic\_id INT UNSIGNED AUTO\_INCREMENT,

name VARCHAR(50) NOT NULL,

PRIMARY KEY(dev\_topic\_id)

) ENGINE=InnoDB";

$newTable\_engineers\_phones = "CREATE TABLE engineers\_phones(

engineer\_id INT UNSIGNED NOT NULL,

phone VARCHAR(20) NOT NULL,

PRIMARY KEY(engineer\_id,phone),

FOREIGN KEY (engineer\_id) REFERENCES engineers (engineer\_id)

ON DELETE CASCADE

ON UPDATE CASCADE

) ENGINE=InnoDB";

$newTable\_grades = "CREATE TABLE grades(

engineer\_id INT UNSIGNED NOT NULL,

project\_id INT UNSIGNED NOT NULL,

grade TINYINT NOT NULL,

month TINYINT NOT NULL,

year SMALLINT NOT NULL,

PRIMARY KEY(engineer\_id,project\_id,month,year),

FOREIGN KEY(engineer\_id) REFERENCES engineers(engineer\_id)

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY(project\_id) REFERENCES projects(project\_id)

ON DELETE CASCADE

ON UPDATE CASCADE

) ENGINE=InnoDB";

$newTable\_project\_participations = "CREATE TABLE project\_participations(

engineer\_id INT UNSIGNED NOT NULL,

project\_id INT UNSIGNED NOT NULL,

PRIMARY KEY(engineer\_id,project\_id),

FOREIGN KEY (engineer\_id) REFERENCES engineers (engineer\_id)

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (project\_id) REFERENCES projects (project\_id)

ON DELETE CASCADE

ON UPDATE CASCADE

) ENGINE=InnoDB";

$newTable\_development\_tools = "CREATE TABLE development\_tools(

auto\_id INT UNSIGNED NOT NULL AUTO\_INCREMENT,

project\_id INT UNSIGNED NOT NULL,

dev\_topic\_id INT UNSIGNED NOT NULL,

tool VARCHAR(50) NOT NULL,

PRIMARY KEY(auto\_id),

FOREIGN KEY (project\_id) REFERENCES projects (project\_id)

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (dev\_topic\_id) REFERENCES development\_topics (dev\_topic\_id)

ON DELETE RESTRICT

ON UPDATE CASCADE

) ENGINE=InnoDB";

if ($conn->query($newTable\_software\_topics)==TRUE)

echo "software\_topic TABLE created successfully" . "<BR>";

if ($conn->query($newTable\_engineers)==TRUE)

echo "engineers TABLE created successfully" . "<BR>";

if ($conn->query($newTable\_projects)==TRUE)

echo "projects TABLE created successfully" . "<BR>";

if ($conn->query($newTable\_milestones)==TRUE)

echo "milestones TABLE created successfully" . "<BR>";

if ($conn->query($newTable\_development\_topics)==TRUE)

echo "development\_topics TABLE created successfully" . "<BR>";

if ($conn->query($newTable\_engineers\_phones)==TRUE)

echo "engnieers\_phones TABLE created successfully" . "<BR>";

if ($conn->query($newTable\_grades)==TRUE)

echo "grades TABLE created successfully" . "<BR>";

if ($conn->query($newTable\_project\_participations)==TRUE)

echo "project\_participations TABLE created successfully" . "<BR>";

if ($conn->query($newTable\_development\_tools)==TRUE)

echo "development\_tools TABLE created successfully" . "<BR>";

}

**function initializeValues($conn)**{

$conn->query("USE projectDB");

//insertSoftwareTopics($conn); //prepered statement

$software\_topics = "INSERT INTO software\_topics (name,specialty) VALUES

('JAVA','FRONTEND'),

('JAVA','BACKEND'),

('JAVA','QA'),

('REACT','FRONTEND'),

('C++', 'FRONTEND'),

('C++', 'BACKEND'),

('C++', 'QA'),

('ANGULAR','FRONTEND'),

('nodeJS','BACKEND'),

('nodeJS','QA')

";

$engineers="INSERT INTO engineers(firstname,lastname,birthdate,address,software\_topic\_id) VALUES

('aa','AA','2000/1/1','street a',1),

('bb','BB','2000/2/2','street b',2),

('cc','CC','2000/3/3','street c',3),

('dd','DD','2000/4/4','street d',4),

('ee','EE','2000/5/5','street e',5),

('ff','FF','2000/6/6','street f',6),

('gg','GG','2000/7/7','street g',6),

('hh','HH','2000/8/8','street h',7),

('ii','II','2000/9/9','street i',8),

('jj','JJ','2000/10/10','street j',8)

";

$engineers\_phones = "INSERT INTO engineers\_phones (engineer\_id,phone) VALUES

(1,'100'),

(1,'1000'),

(2,'200'),

(3,'300'),

(3,'3000'),

(4,'400'),

(5,'500'),

(6,'600'),

(8,'800'),

(9,'900'),

(9,'9000')

";

$projects = "INSERT INTO projects (projectName,clientName,startDate) VALUES

('P1','C1','2017/1/1'),

('P2','C2','2017/2/2'),

('P3','C3','2017/3/3'),

('P4','C4','2017/4/4'),

('P5','C5','2017/5/5'),

('P6','C6','2017/6/6'),

('P7','C7','2017/7/7'),

('P8','C8','2017/8/8'),

('P9','C9','2017/9/9'),

('P10','C10','2017/10/10')

";

$milestones = "INSERT INTO milestones (project\_id, amount, targetDate, description) VALUES

(1, 1200, '2018/01/20', 'M1'),

(1, 3300, '2018/01/25', 'M2'),

(1, 990, '2018/02/06', 'M3'),

(2, 840, '2018/01/18', 'M1'),

(2, 6200, '2018/02/24', 'M2'),

(2, 1200, '2018/03/20', 'M3'),

(3, 3100, '2018/01/24', 'M1'),

(3, 4200, '2018/03/20', 'M2'),

(5, 1200, '2018/01/20', 'M1'),

(5, 1200, '2018/02/20', 'M2')

";

$project\_participations = "INSERT INTO project\_participations (engineer\_id, project\_id) VALUES

(1, 1), (1, 3), (1, 4),

(2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (2, 9), (2, 10),

(3, 3), (3, 5), (3, 7), (3,9),

(4, 2), (4, 4), (4, 6), (4, 8), (4, 10),

(9, 1), (9, 2), (9, 3), (9, 4), (9, 5), (9, 6), (9, 7), (9, 8), (9, 9), (9, 10)

";

$development\_topics = "INSERT INTO development\_topics (name) VALUES

('Configuration Management'),

('Design'),

('Requirements management'),

('Task management'),

('Encoding'),

('Decoding'),

('QA')

";

$development\_tools = "INSERT INTO development\_tools (project\_id, dev\_topic\_id, tool) VALUES

(1,1,'GIT'),

(2,1,'GIT'),

(3,1,'GIT'),

(1,2,'Photoshop'),

(4,2,'Illustrator'),

(6,2,'Paint'),

(8,2,'Photoshop'),

(9,2,'Illustrator'),

(5,1,'GitHUB'),

(8,1,'GitBucket')

";

$grades = "INSERT INTO grades (engineer\_id, project\_id, month, year, grade) VALUES

(2, 1, 1, 2000, 9),(2, 2, 1, 2000, 6),(2, 3, 1, 2000, 5),(2, 4, 1, 2000, 2),(2, 5, 1, 2000, 8),

(2, 6, 1, 2000, 6),(2, 7, 1, 2000, 9),(2, 8, 1, 2000, 7),(2, 9, 1, 2000, 3),(2, 10, 1, 2000, 10),

(9, 1, 1, 2000, 8),(9, 2, 1, 2000, 7),(9, 3, 1, 2000, 9),(9, 4, 1, 2000, 3),(9, 5, 1, 2000, 8),

(9, 6, 1, 2000, 8),(9, 7, 1, 2000, 9),(9, 8, 1, 2000, 2),(9, 9, 1, 2000, 7),(9, 10, 1, 2000, 9),

(4, 2, 1, 2000, 7),(4, 4, 1, 2000, 5),(4, 6, 1, 2000, 6),(4, 8, 1, 2000, 5),(4, 10, 1, 2000, 9)

";

try{

$conn->autocommit(FALSE);

echo "<BR><BR>";

$conn->query($software\_topics);

$conn->query($engineers);

$conn->query($engineers\_phones);

$conn->query($projects);

$conn->query($project\_participations); //e2 and e9 participating in all projects

$conn->query($milestones);

$conn->query($development\_topics);

$conn->query($development\_tools);

$conn->query($grades);

$conn->commit();

echo "software\_topics VALUES inserted successfully" . "<BR>";

echo "engineers VALUES inserted successfully" . "<BR>";

echo "engineers\_phones VALUES inserted successfully" . "<BR>";

echo "projects VALUES inserted successfully" . "<BR>";

echo "milestones VALUES inserted successfully" . "<BR>";

echo "development\_topics VALUES inserted successfully" . "<BR>";

echo "development\_tools VALUES inserted successfully" . "<BR>";

echo "grades VALUES inserted successfully" . "<BR>";

echo "<BR><BR>";

echo "<div style='margin:0 auto;font-size:16px;font-weight:bold;'>

<a href='../GUI'>לחץ כאן להיכנס לממשק</a>

</div>

";

}

catch (Exception $e) {

$conn->rollback();

echo "error: insert has failed";

}

}

/\*

function insertSoftwareTopics($conn){ //for prepared statement

$stmt = $conn->prepare("INSERT INTO software\_topics (name, specialty) VALUES (?, ?)");

$stmt->bind\_param("ss", $name, $specialty);

// set parameters and execute

$name = 'JAVA';

$specialty = 'FRONTEND';

$stmt->execute();

$name = 'JAVA';

$specialty = 'BACKEND';

$stmt->execute();

$name = 'JAVA';

$specialty = 'QA';

$stmt->execute();

$name = 'ANGULAR';

$specialty = 'FRONTEND';

$stmt->execute();

echo "New records created successfully";

$stmt->close();

}

\*/

?>

# Project Backend Source Code:

Modify Data(ModifyDB/filename): (edit/insert entity)

**ModifyDB/EditEngineer.php**

<?php

$rowID = $\_POST["rowID"]; //edit-new flag

$firstname = $\_POST["firstname"];

$lastname = $\_POST["lastname"];

$topic\_id = $\_POST["topic\_id"];

$birthdate = $\_POST["birthdate"];

$address = $\_POST["address"];

$phones = $\_POST["phones"];

// Create connection

$conn = new mysqli($servername, $username, $password,$db);

// Check connection

if ($conn->connect\_error) {

echo "error: " . $conn->connect\_error;

exit();

}

try{

$conn->autocommit(FALSE);

if(is\_null($rowID) || $rowID==''){ //add new engineer

$sql = "INSERT INTO engineers (software\_topic\_id, firstname, lastname, birthdate,address)

VALUES (**$topic\_id**,'**$firstname**','**$lastname**','**$birthdate**','**$address**')";

$conn->query($sql);

$new\_id = $conn->insert\_id;

foreach($phones as $phone) {

$sql = "INSERT INTO engineers\_phones (engineer\_id,phone) VALUES (**$new\_id**,'**$phone**')";

$conn->query($sql);

}

echo "**$new\_id**";

} else { //edit engineer

$sql = "UPDATE engineers

SET software\_topic\_id='**$topic\_id**', firstname = '**$firstname**', lastname = '**$lastname**', birthdate='**$birthdate**', address='**$address**'

WHERE engineer\_id = **$rowID**

";

$conn->query($sql);

//delete engineer\_phones and add it from scratch

$sql = "DELETE FROM engineers\_phones WHERE engineer\_id=**$rowID**";

$conn->query($sql);

foreach($phones as $phone) {

$sql = "INSERT INTO engineers\_phones (engineer\_id,phone) VALUES (**$rowID**,'**$phone**')";

$conn->query($sql);

}

}

$conn->commit();

} catch (Exception $e) {

$conn->rollback();

echo "error: insert has failed";

}

$conn->close;

?>

**ModifyDB/EditProject.php**

<?php

// Create connection

$conn = new mysqli($servername, $username, $password,$db);

// Check connection

if ($conn->connect\_error)

die("error: " . $conn->connect\_error);

if($rowID==0){ //add new topic

$sql = "INSERT INTO projects (projectName, clientName, startDate)

VALUES ('**$projectName**','**$clientName**', '**$startDate**')";

if ($conn->query($sql) === FALSE)

die("error: **$conn->error**");

$new\_id = $conn->insert\_id;

echo "**$new\_id**";

} else { //edit topic

$sql = "UPDATE projects

SET projectName = '**$projectName**', clientName = '**$clientName**', startDate='**$startDate**'

WHERE project\_id = **$rowID**

";

if($conn->query($sql)==FALSE)

die("error: **$conn->error**");

}

$conn->close;

?>

**ModifyDB/EditMilestone.php**

<?php

$conn = new mysqli($servername, $username, $password,$db);

// Check connection

if ($conn->connect\_error)

die("error: " . $conn->connect\_error);

if($rowID==0){ //add new topic

$sql = "INSERT INTO milestones (project\_id, amount, targetDate, description)

VALUES (**$projectID**, **$amount**, '**$targetDate**', '**$desc**')";

if ($conn->query($sql) === FALSE)

die("error: **$conn->error**");

$new\_id = $conn->insert\_id;

echo "**$new\_id**";

} else { //edit topic

$sql = "UPDATE milestones

SET description = '**$desc**', targetDate='**$targetDate**', amount=**$amount**

WHERE milestone\_id = **$rowID**

";

if($conn->query($sql)==FALSE)

die("error: **$conn->error**");

}

$conn->close;

?>

**ModifyDB/EditSoftwareTopic.php:**

$conn = new mysqli($servername, $username, $password,$db);

// Check connection

if ($conn->connect\_error)

die("error: " . $conn->connect\_error);

/\*check if topic is already exists\*/

$sql = "SELECT \* FROM software\_topics WHERE name='**$name**' AND specialty='**$specialty**' AND topic\_id<>**$rowID**";

$result = $conn->query($sql);

if($result->num\_rows!=0){

echo "alreadyExists";

exit();

}

if($rowID==0){ //add new topic

$sql = "INSERT INTO software\_topics (name, specialty)

VALUES ('**$name**','**$specialty**')";

if ($conn->query($sql) === FALSE)

die("error: **$conn->error**");

$new\_id = $conn->insert\_id;

echo "**$new\_id**";

} else { //edit topic

$sql = "UPDATE software\_topics

SET name = '**$name**', specialty = '**$specialty**'

WHERE topic\_id = **$rowID**

";

if($conn->query($sql)==FALSE)

die("error: **$conn->error**");

}

$conn->close;

**ModifyDB/EditDevelopmentTopic.php:**

$conn = new mysqli($servername, $username, $password,$db);

if($rowID==0){ //add new topic

$sql = "INSERT INTO development\_topics (name)

VALUES ('**$name**')";

if ($conn->query($sql) === FALSE)

die("error: **$conn->error**");

$new\_id = $conn->insert\_id;

echo "**$new\_id**";

} else { //edit topic

$sql = "UPDATE development\_topics

SET name = '**$name**'

WHERE dev\_topic\_id = **$rowID**

";

if($conn->query($sql)==FALSE)

die("error: **$conn->error**");

}

$conn->close;

**ModifyDB/EditTool.php**

if($rowID==0){ //add new topic

$sql = "INSERT INTO development\_tools (project\_id, dev\_topic\_id, tool)

VALUES (**$project\_id**, **$topic\_id**, '**$tool**')";

if ($conn->query($sql) === FALSE)

die("error: **$conn->error**");

$new\_id = $conn->insert\_id;

echo "**$new\_id**";

} else { //edit topic

$sql = "UPDATE development\_tools

SET dev\_topic\_id=**$topic\_id**, tool='**$tool**'

WHERE auto\_id = **$rowID**

";

if($conn->query($sql)==FALSE)

die("error: **$conn->error**");

}

**ModifyDB/RankProject.php:**

$sql = "INSERT INTO grades (engineer\_id, project\_id, grade, month, year)

VALUES (**$engineer\_id**,**$project\_id**, **$grade**, **$month**, **$year**)";

**ModifyDB/AddEngineerToProject.php**

$sql = "INSERT INTO project\_participations (project\_id, engineer\_id)

VALUES (**$projectID**, **$engineerID**)";

if($conn->query($sql)==FALSE)

die("error. cannot insert entity");

**modifyDB/DeleteEngineerFromProject.php**

$sql = "DELETE FROM project\_participations WHERE engineer\_id='**$engineerID**' AND project\_id='**$projectID**'";

if($conn->query($sql)==FALSE)

die("error. cannot delete entity");

**modifyDB/DeleteDynamicRow.php:** (delete entity from dynamic table with dynamic params).

$sql = "DELETE FROM **$tableName** WHERE **$fieldName**=**$fieldVal**";

if($conn->query($sql)==FALSE)

die("error. cannot delete entity");

**Select Data pages:** The application isolates the view from the data. Javascript will make ajax call to a

server-side-page.php and will get the data wrapped by JSON format. Then, javascript will display this data in bootstrap table.

**jsonsDB/Enginners.php**

if($mode=='occupiedAllProjects'){

$sql = "

SELECT a.engineer\_id, e.firstname, e.lastname, e.address, e.birthdate,

TIMESTAMPDIFF(YEAR,e.birthdate,NOW()) AS age

, GROUP\_CONCAT(distinct p.phone SEPARATOR '<BR>') AS phones,

s.name AS topicName, s.specialty AS topicSpecialty, s.topic\_id

FROM project\_participations a

LEFT JOIN engineers\_phones p ON a.engineer\_id=p.engineer\_id

LEFT JOIN engineers e ON e.engineer\_id=a.engineer\_id

LEFT JOIN software\_topics s ON s.topic\_id=e.software\_topic\_id

WHERE a.engineer\_id IN

(SELECT b.engineer\_id FROM project\_participations b GROUP BY b.engineer\_id

HAVING COUNT(b.engineer\_id)=(SELECT COUNT(project\_id) FROM projects))

GROUP BY engineer\_id

";

} else {

$sql = "

SELECT a.engineer\_id, a.firstname, a.lastname, a.address, a.birthdate,

TIMESTAMPDIFF(YEAR,a.birthdate,NOW()) AS age,

GROUP\_CONCAT(distinct b.phone SEPARATOR '<br>') AS phones,

c.name AS topicName, c.specialty AS topicSpecialty, c.topic\_id

FROM engineers a

LEFT JOIN engineers\_phones b ON a.engineer\_id=b.engineer\_id

LEFT JOIN software\_topics c ON a.software\_topic\_id=c.topic\_id

GROUP BY a.engineer\_id

";

}

$result = $conn->query($sql);

$jsonArr = array();

if ($result->num\_rows > 0) {

while($rs = $result->fetch\_assoc()) {

$jsonArr[] = array(

'rowID' => $rs['engineer\_id'],

'firstname' => $rs['firstname'],

'lastname' => $rs['lastname'],

'age' => $rs['age'],

'address' => $rs['address'],

'phones' => $rs['phones'],

'topicName' => $rs['topicName'] . " - " . $rs['topicSpecialty'],

'topic\_id' => $rs['topic\_id'],

'birthdate' => $rs['birthdate'],

'rank' => "<span class='glyphicon glyphicon-signal'></span>"

);

}

}

echo json\_encode($jsonArr);

**JsonsDB/projects.php**

if(is\_null($engineer\_id) || $engineer\_id==''){ //return all projects data

$sql = "SELECT \*, DATE\_FORMAT(startDate, '%d/%m/%Y') as startDateStr FROM projects";

} else { //engineer's projects

$sql = "

SELECT p.projectName, p.project\_id

FROM project\_participations a

LEFT JOIN projects p ON a.project\_id=p.project\_id

WHERE engineer\_id=**$engineer\_id**

";

}

**JsonsDB/SoftwareTopics.php**

$sql = "SELECT \* FROM software\_topics order by name ASC";

**JsonsDB/milestones.php**

$sql = "SELECT \*, DATE\_FORMAT(a.targetDate, '%d/%m/%Y') as targetDateStr, b.projectName

FROM milestones a

LEFT JOIN projects b ON a.project\_id = b.project\_id

WHERE targetDate<DATE\_ADD(CURDATE(), INTERVAL 30 DAY) AND targetDate>=CURDATE()

ORDER BY a.project\_id ASC

";

**JsonsDB/milestonesPerProject.php**

$sql = "SELECT \*, DATE\_FORMAT(a.targetDate, '%d/%m/%Y') as targetDateStr

FROM milestones a

WHERE project\_id=**$projectID**";

**JsonsDB/engineersPerProject.php**

$sql = "SELECT a.engineer\_id, b.firstname, b.lastname , c.name AS topicName, c.specialty AS topicSpecialty

FROM project\_participations a

LEFT JOIN engineers b ON a.engineer\_id=b.engineer\_id

LEFT JOIN software\_topics c ON b.software\_topic\_id=c.topic\_id

WHERE project\_id=**$projectID**

order by c.topic\_id ASC

";

if($isGroup){ /\*groupBy\*/

$sql = "

SELECT a.engineer\_id, c.name AS topicName, c.specialty AS topicSpecialty,

b.firstname, b.lastname

FROM project\_participations a

LEFT JOIN engineers b ON a.engineer\_id=b.engineer\_id

LEFT JOIN software\_topics c ON b.software\_topic\_id=c.topic\_id

WHERE project\_id=**$projectID**

GROUP BY c.topic\_id

order by c.topic\_id ASC

";

}

**JsonsDB/developmentTopics.php**

$sql = "SELECT \* FROM development\_topics order by name ASC";

**JsonsDB/developmentToolsPerTopic.php**

$sql = "SELECT a.\*, b.projectName

from development\_tools a

LEFT JOIN projects b ON a.project\_id=b.project\_id

WHERE dev\_topic\_id=**$devID**

";

**jsonsDB/DevelopmentToolsPerProject.php**

$sql = "SELECT a.\*, b.name AS topicName, b.dev\_topic\_id AS topic\_id

from development\_tools a

LEFT JOIN development\_topics b ON a.dev\_topic\_id=b.dev\_topic\_id

WHERE project\_id=**$projectID**

ORDER BY b.name

";

**JsonsDB/projectsRanks.php**

$sql = "SELECT \*, ROUND(AVG(g.grade),1) as avgGrade, p.projectName

FROM grades g

INNER JOIN projects p ON p.project\_id=g.project\_id

GROUP BY g.project\_id

ORDER BY avgGrade DESC

";