1.

```
$: sudo apt install mariadb-server
$: sudo mariadb
```

```
create database library;
use library;
MariaDB [(none)]> create table if not exists books (b_id INT PRIMARY KEY AUTO_IN
CREMENT, name VARCHAR(100) NOT NULL);
MariaDB [(none)]> create table if not exists orders (o_id INT PRIMARY KEY AUTO_I
NCREMENT, b_id INT REFERENCES books(b_id), address VARCHAR(100) NOT NULL, status
 BOOLEAN);
insert into books values(1,'name1');
insert into books values(2,'name2');
MariaDB [(none)]> grant all privileges on *.* to 'root'@'localhost' identified b
y 'password';
flush privileges;
```

…/lab4/add_main.php

```
<html>
        <head><title>Add orders</title></head>
        <body>
                <?php
                        function addToDB($_id, $_addr, &$_error){
                                try{
                                        $conn=mysqli_connect("localhost","root","password");
                                        mysqli_select_db($conn,"library");
                                        $query="INSERT INTO orders values(NULL,$_id,'$_addr',fal
se);";
                                        mysqli_query($conn,$query);
                                        mysqli_close($conn);
                                }
                                catch(mysqli_sql_exception $exp){
                                        $_error=$exp->getMessage();
                                        return false;
                                }
                                return true;
                        }

                        try{
                                $conn=mysqli_connect("localhost","root","password");
                                mysqli_select_db($conn,"library");
                                $result=mysqli_query($conn,"select * from books;");
                                echo "<p>Book name: </p>";
                                echo "<form action='{$_SERVER['PHP_SELF']}' method='post'>";
                                echo "<select name='selection'>";
                                while($row=mysqli_fetch_row($result)){
                                        echo "<option value='$row[0]'>$row[1]</option>";
                                }
                                echo "</select><br>";
                                echo "<input type='text' name='addr'><br>";
                                echo "<input type='submit' value='Add order'>";
                                echo "</form>";
                                mysqli_close($conn);
                        }
                        catch(mysqli_sql_exception $exp){
                                die("Error: ".$exp->getMessage());
                        }

                        if(isset($_POST['selection']) && isset($_POST['addr'])){
                                $db_error="";
                                $ret=addToDB($_POST['selection'],$_POST['addr'],$db_error);
                                if(!$ret) print "Error: $db_error";
                        }

                        echo "<a href='show_main.php'>Show status<br></a>";
                ?>
        </body>
</html>
```

…/lab4/show_main.php

```
<html>
        <head><title>Show orders</title></head>
        <body>
                <?php
                        try{
                                $conn=mysqli_connect("localhost","root","password");
                                mysqli_select_db($conn,"library");
                                $result=mysqli_query($conn,"select * from orders");
                                echo "<p>Orders:</p>";
                                echo "<table border='1' cellpadding='0px'>";
                                echo "<tr>";
                                echo "<td width='14px' height='15px' size=1>#</td>";
                                echo "<td width='14px' height='15px' size=1>Address</td>";
                                echo "<td width='14px' height='15px' size=1>Status</td>";
                                echo "</tr>";
                                while($row=mysqli_fetch_row($result)){
                                        echo "<tr>";
                                        echo "<td width='14px' height='15px' size=1>$row[0]</td>
";
                                        echo "<td width='14px' height='15px' size=1>$row[2]</td>
";
                                        echo "<td width='14px' height='15px' size=1>$row[3]</td>
";
                                        echo "</tr>";
                                }
                                echo "</table>";
                                mysqli_close($conn);
                        }
                        catch(mysqli_sql_exception $exp){
                                die("Error: ".$exp->getMessage());
                        }

                        echo "<a href='add_main.php'>Add orders</a>";
                ?>
        </body>
</html>
```

```
$: sudo apt install php
…/lab4/$: php –server 127.0.0.1:8080
```

2. [1]

```
$: sudo apt install nginx
$: sudo apt install php-fpm
$: mysqldump -u root -p –databases library > /tmp/mydb_dump.sql
$: scp /tmp/mydb_dump.sql user@192.168.141.16:/tmp
$: scp -r /home/user/lab4 user@192.168.141.16:/home/user
```

[2]

```
$: sudo apt install openssh-server
$: sudo apt install php
$: sudo apt install mariadb-server
$: sudo apt install nginx
$: sudo apt install php-fpm
$: sudo mariadb -u root -p < /tmp/mydb_dump.sql
$: sudo apt install nginx
$: sudo apt install php-fpm
```

3. [1]
/etc/mysql/mariadb.conf.d/50-server.cnf

```
bind-address            = 192.168.141.15
server-id               = 1
replicate-do-db         = library
binlog-do-db            = library

#
# * Fine Tuning
#

#key_buffer_size        = 128M
#max_allowed_packet     = 1G
#thread_stack           = 192K
#thread_cache_size      = 8
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
#myisam_recover_options = BACKUP
#max_connections        = 100
#table_cache            = 64

#
# * Logging and Replication
#

# Note: The configured log file or its directory need to be created
# and be writable by the mysql user, e.g.:
# $ sudo mkdir -m 2750 /var/log/mysql
# $ sudo chown mysql /var/log/mysql

# Both location gets rotated by the cronjob.
# Be aware that this log type is a performance killer.
# Recommend only changing this at runtime for short testing periods if needed!
general_log_file       = /var/log/mysql/mysql.log
general_log            = 1

# When running under systemd, error logging goes via stdout/stderr to journald
# and when running legacy init error logging goes to syslog due to
# /etc/mysql/conf.d/mariadb.conf.d/50-mysqld_safe.cnf
# Enable this if you want to have error logging into a separate file
log_error = /var/log/mysql/error.log
# Enable the slow query log to see queries with especially long duration
#log_slow_query_file    = /var/log/mysql/mariadb-slow.log
#log_slow_query_time    = 10
#log_slow_verbosity     = query_plan,explain
#log-queries-not-using-indexes
#log_slow_min_examined_row_limit = 1000

# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replica, see README.Debian about other
#       settings you may need to change.
#server-id              = 1
log_bin                = /var/log/mysql/mysql-bin.log
expire_logs_days       = 10
```

```
sudo mkdir /var/log/mysql
```

```
sudo chown mysql:mysql /var/log/mysql
```

```
MariaDB [(none)]> grant replication slave on *.* to "user"@"192.168.141.16" iden
tified by "password";
```

```
MariaDB [(none)]> show master status\G;
*************************** 1. row ***************************
            File: mysql-bin.000001
        Position: 328
    Binlog_Do_DB: library
Binlog_Ignore_DB:
1 row in set (0,000 sec)
```

```
MariaDB [(none)]> change master to master_host = "192.168.141.16", master_port =
 3306, master_user = "user", master_password = "password", master_log_file = "my
sql-bin.000001", master_log_pos = 328;
```

```
MariaDB [(none)]> start slave;
```

```
user@user:~/lab4$ php --server 192.168.141.15:8081
```

[2]

/etc/mysql/mariadb.conf.d/50-server.cnf

```
bind-address            = 192.168.141.16
server-id               = 2
replicate-do-db         = library
binlog-do-db            = library

#
# * Fine Tuning
#

#key_buffer_size        = 128M
#max_allowed_packet     = 1G
#thread_stack           = 192K
#thread_cache_size      = 8
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
#myisam_recover_options = BACKUP
#max_connections        = 100
#table_cache            = 64

#
# * Logging and Replication
#

# Note: The configured log file or its directory need to be created
# and be writable by the mysql user, e.g.:
# $ sudo mkdir -m 2750 /var/log/mysql
# $ sudo chown mysql /var/log/mysql

# Both location gets rotated by the cronjob.
# Be aware that this log type is a performance killer.
# Recommend only changing this at runtime for short testing periods if needed!
general_log_file       = /var/log/mysql/mysql.log
general_log            = 1

# When running under systemd, error logging goes via stdout/stderr to journald
# and when running legacy init error logging goes to syslog due to
# /etc/mysql/conf.d/mariadb.conf.d/50-mysqld_safe.cnf
# Enable this if you want to have error logging into a separate file
log_error = /var/log/mysql/error.log
# Enable the slow query log to see queries with especially long duration
#log_slow_query_file    = /var/log/mysql/mariadb-slow.log
#log_slow_query_time    = 10
#log_slow_verbosity     = query_plan,explain
#log-queries-not-using-indexes
#log_slow_min_examined_row_limit = 1000

# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replica, see README.Debian about other
#       settings you may need to change.
#server-id              = 1
log_bin                = /var/log/mysql/mysql-bin.log
expire_logs_days       = 10
```

```
sudo mkdir /var/log/mysql
sudo chown mysql:mysql /var/log/mysql
```

```
MariaDB [(none)]> grant replication slave on *.* to "user"@"192.168.141.15" identified by "p
assword";
```

```
MariaDB [(none)]> show master status\G;
*************************** 1. row ***************************
            File: mysql-bin.000001
        Position: 328
    Binlog_Do_DB: library
Binlog_Ignore_DB:
1 row in set (0,019 sec)
```

```
MariaDB [(none)]> change master to master_host = "192.168.141.15", master_port = 3306, maste
r_user = "user", master_password = "password", master_log_file = "mysql-bin.000001", master_
log_pos = 328;
```

```
MariaDB [(none)]> start slave;
```

/etc/nginx/sites-available/default

```
upstream application{
        server 192.168.141.15:8081;
        server 192.168.141.16:8081;
}

server {
        listen 8080;
        listen [::]:8080;

        server_name myserver.local www.myserver.local;

        # SSL configuration
        #
        # listen 443 ssl default_server;
        # listen [::]:443 ssl default_server;
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
        #
        # Read up on ssl_ciphers to ensure a secure configuration.
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

        #root /var/www/html;
        root /home/user/lab4;

        # Add index.php to the list if you are using PHP
        #index index.html index.htm index.nginx-debian.html;
        index add_main.php show_main.php;

        server_name myserver.local;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                #try_files $uri $uri/ =404;
                proxy_pass http://application;
        }

        # pass PHP scripts to FastCGI server
        #
        #location ~ \.php$ {
        #       include snippets/fastcgi-php.conf;
        #
        #       # With php-fpm (or other unix sockets):
        #       fastcgi_pass unix:/run/php/php7.4-fpm.sock;
        #       # With php-cgi (or other tcp sockets):
        #       fastcgi_pass 127.0.0.1:9000;
        #}
```

$: sudo apt install bind9

/etc/bind/named.conf.local

```
//
// Do any local configuration here
//

zone "myserver.local"{
        type master;
        file "/etc/bind/db.myserver.local";
};
```

/etc/bind/db.myserver.local

```
$TTL 86400;

@ IN SOA dns2.myserver.local. root.myserver.local. (
        2022103007 ; Serial
        600 ; Refresh
        3600 ; Retry
        1w ; Expirt
        360) ; Negative cache TTL

@ IN NS dns2.myserver.local.
@ IN A 192.168.141.16
dns2 IN A 192.168.141.16
www IN A 192.168.141.16
```

```
user@user:~/lab4$ php --server 192.168.141.16:8081
```

[3]

/etc/systemd/resolved.conf

```
[Resolve]
# Some examples of DNS serv
# Cloudflare: 1.1.1.1#cloud
# Google:     8.8.8.8#dns.g
# Quad9:      9.9.9.9#dns.q
DNS=192.168.141.16
#FallbackDNS=
Domains=myserver.local
```

/etc/nsswitch.conf

```
hosts:          files mdns4_minimal dns
```

$: sudo systemctl stop system-resolved $: sudo systemctl restart system-resolved

В браузере – https://www.myserver.local:8081/add_main.php