

Introduzione

Breve introduzione sull'azienda

HIGHTEL TOWERS è una Tower Company e Operatore di Comunicazioni Elettroniche dal 2001, iscritta al Registro degli Operatori di Comunicazione presso l'Autorità per le Garanzie nelle Comunicazioni.

Il suo core business si articola lungo due linee differenti di business:

1- la progettazione e realizzazione di torri per telecomunicazioni

2- la fornitura di servizi agli operatori di telefonia mobile

La mission aziendale è offrire servizi di Housing e Towering per impianti TLC, servizi di consulenza, ingegneria, progettazione, realizzazione e gestione di infrastrutture di rete.

Inoltre, sempre nella mission, rientra la gestione di contratti di locazione e di gestione di patrimoni immobiliari e industriali.



Breve introduzione del software progettato

Il software oggetto della presente tesina è stato progettato con l'obiettivo di raccogliere e gestire tutte le informazioni inerenti le attività che si articolano nel processo di Housing offerto ai diversi gestori di comunicazioni elettroniche.

Nel software si identificano sostanzialmente due soggetti:

1. Amministratore (Hightel Towers), che ha il controllo dei contratti alla locazione degli spazi concessi ai gestori, su tutte le stazioni radiobase (SRB)
2. User (SRB), che si identifica con ogni singola stazione radiobase e nel quale sono archiviati tutti i contratti stipulati con i gestori per l'occupazione degli spazi su quella determinata SRB.

Il software è in fase di sviluppo per la gestione dell'asset dell'Est Europa, Albania, Macedonia e Kosovo.

Organizzazione del software.

Il software si sviluppa su un database in SQL (Structured Query Language).

Inoltre, per creare la pagina web di interfaccia sono stati usati i seguenti strumenti:

1. Java Script
2. Sql
3. Xamp
4. Notepad ++
5. Models
6. Html
7. Css
8. Bootstrap
9. Php

Lo schema base del programma è così semplificato:

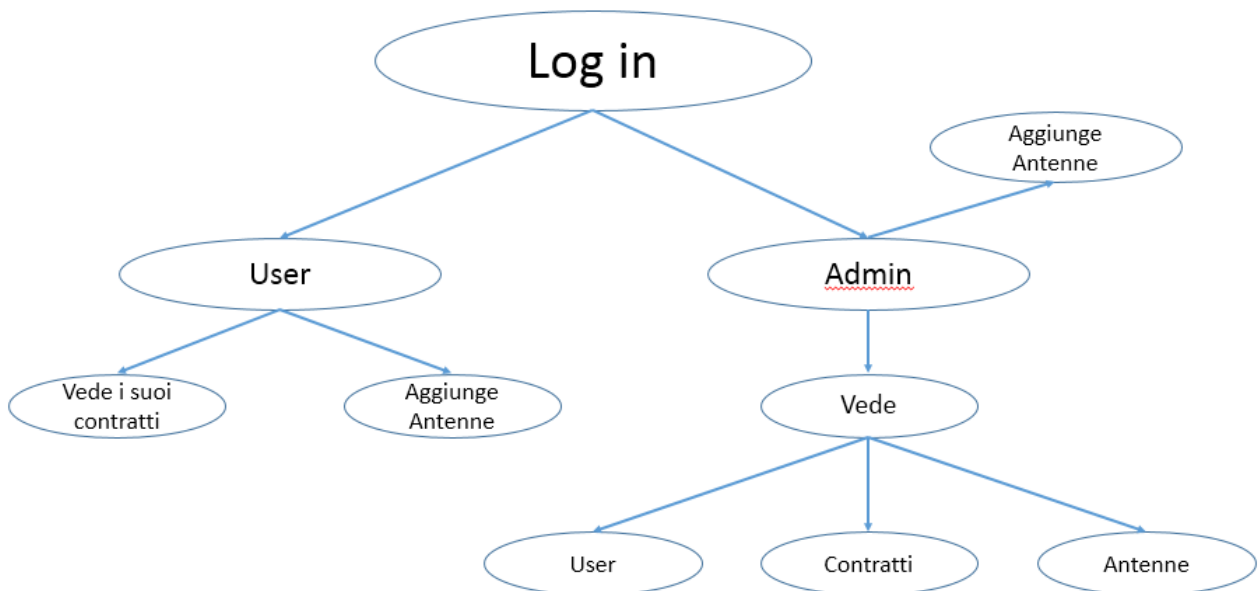


Fig1.0

Login: Pagina principale web divisa in

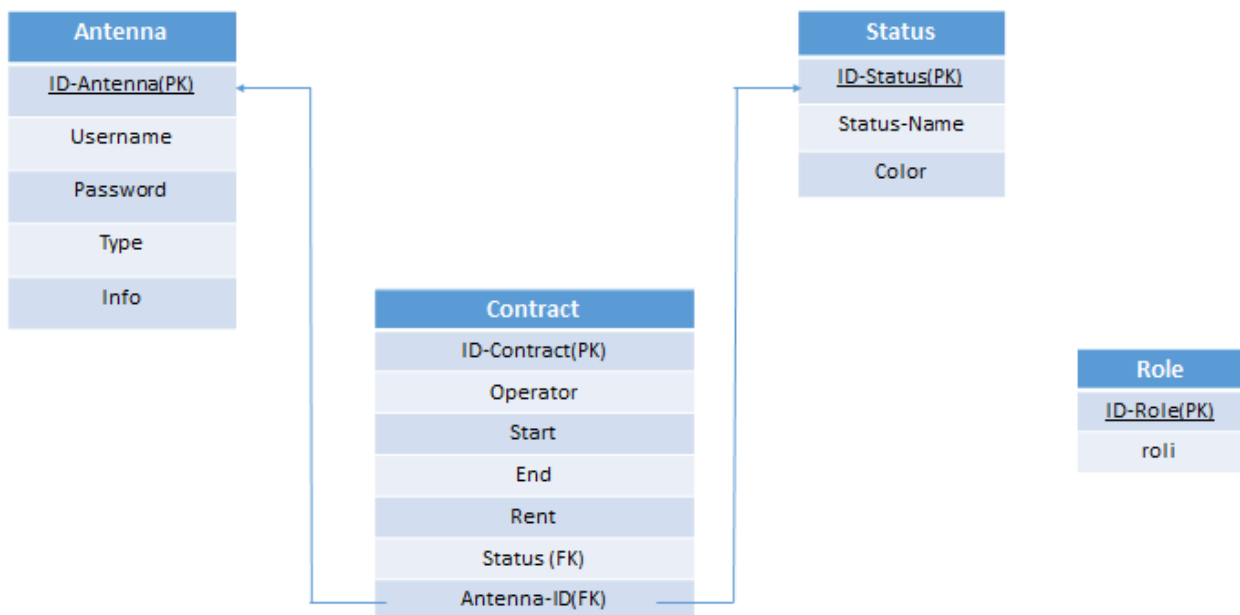
1. User
2. Administrator

Come precedentemente evidenziato:

User è sostanzialmente ciascuna SRB. Attraverso l'accesso di User si può vedere e modificare solo i dati riferiti a quella specifica SRB, comprendenti i contratti e le diverse info in relazione alle antenne.

Administrator è invece il gestore della pagina web, che può monitorare tutto l'asset, aggiungere nuove utenze, vedere tutti gli user presenti nel database, tutti i contratti e le antenne.

Nel dettaglio, segue una schema relazionale delle tabelle presenti nel database:



PK-Primary Key FK-Foreign Key (puntano a un PK)

Definizione Primary Key: tabella che include in genere una colonna o una combinazione di colonne i cui valori identificano in modo univoco ogni riga della tabella. Queste colonne sono denominate chiave primaria e garantiscono l'integrità di entità della tabella. Per creare una chiave primaria, è possibile definire un vincolo PRIMARY KEY in fase di creazione o di modifica di una tabella.

Definizione Foreign Key: nel contesto dei database relazionali, è un vincolo di integrità referenziale tra due o più tabelle. Essa identifica una o più colonne di una tabella (referenziante) che riferenzia una o più colonne di un'altra tabella (referenziata).

Funzionamento del software e spiegazione dettagliata delle tabelle usate in sql:

Partiamo con la prima pagina php lidhjaDB.php:

Questa pagina permette la connessione del DB con tutto il programma. Per la connessione servono 4 parametri che serviranno per ogni movimento che si effettua nel database.

```
vensi.sql | lidhjaDB.php
1 <?php
2 $servername = "localhost";
3 $username = "root";
4 $password = "";
5 $dbname = "vensi";
6
7 // Create connection
8 $conn = mysqli_connect($servername, $username, $password, $dbname); //alla connessione con db servono questi quattro parametri
9 ?>
```

Login.php: E la pagina principale del programma. (vedasi figura 1.0) che permette l'interfaccia con i due differenti utenti sopra evidenziati (Administrator e User) .

A seguire lo script del programma e le descrizioni:

```
login.php
1 <?php
2 session_start();
3 $email=$_POST["user"]; // serve per prendere l'username dell user
4 $pass= $_POST['pass']; // serve per prendere la password
5 include "lidhjaDB.php"; //
6 if (!$conn) {
7     die("Connection failed: " . mysqli_connect_error()); //Serve per vedere se il collegamento e fatto correttamente
8 }
9
10 $sql = "SELECT * FROM antenna
11     WHERE username='$email' AND password='$pass'"; // questa e la query che prende il username e paswword dalla tabella antenne
12 $result = mysqli_query($conn, $sql); // Serve per prendere i risultati(dati) dal database
13 // output data of each row
14 if (mysqli_num_rows($result) > 0) {
15     while($row = mysqli_fetch_assoc($result)) { //serve per prendere i risultati su un vettore associato
16         if($row['id_antena']!=0) { //serve per Vedere se il tipo d'utente e un user
17             $antenna=$row['id_antena'];
18             $_SESSION["antenna"]=$antenna;
19             $_SESSION["email"]=$email;
20             header('Refresh: 0; URL = http:\\Projekt2\User\User.php'); //serve per portarti sulla pagina user
21         }
22         else if($row['id_antena']==0){ //serve per Vedere se il tipo d'utente e un admin
23             $_SESSION["email"]=$email;
24             header('Refresh: 0; URL = http:\\Projekt2\Administrator\Administrator.php'); //serve per portarti sulla pagina admin
25         }
26     }
27 }
28 else{ // se non si trova nessun antenna con questo username e password
29     echo "Antenna doesnt exist";
30     header('Refresh: 2; URL = http:\\Projekt2\Fillimi.html');
31 }
32 mysqli_close($conn);
```

Altri chiarimenti :

1. \$row = prende tutti i risultati della tabella in base alla colonna, dei vettori associati.
2. \$session = servono per passare i variabili da una pagina web al altra
3. Se antenna id=0 abbiamo un user tipo amministratore se antenna id ≠0 abbiamo a che fare con un user normale
4. Nella riga 17 dello script si fa il confronto per vedere il tipo del user collegato.

Fillimi.html è la schermata della pagina dove possiamo fare il login

```
1 <html>
2 <head>
3 <title>Pika Shitje</title>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
7 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
8 <link rel="stylesheet" type="text/css" href="http://localhost/Projekt2/css/style.css">
9 </head>
10 <body>
11 <div class="container">
12 <div class="card card-container">
13 
14 <p style="color:black;" id="profile-name" class="profile-name-card" >Welcome</p><br>
15 <form method="post" action="http://localhost/Projekt2/login.php" class="form-signin" role="form">
16 <input type="text" id="user" name="user" class="form-control" placeholder="Username" required autofocus>
17 <input type="password" id="pass" name="pass" class="form-control" placeholder="Password" required>
18 <button type="submit" class="btn btn-lg btn-success btn-block btn-signin">Log in <span class="glyphicon glyphicon-ok"></span></button>
19 </form>
20
21
22 </div><!-- /card-container -->
23 </div><!-- /container -->
24 </body>
25 </html>
```

form method="post"- serve per trasmettere i dati in un'altra pagina che viene definita da action ed elaborarli.

Per ogni input serve poi un *id* o un *name* in modo da identificare univocamente quanto inviato alle pagine che gestiscono l'elaborazione i dati.

Praticamente questo è il codice che fa vedere la pagina WEB del login.

Tutti i dati vengono elaborati alla pagina login.php dove si identifica tipo di user.

È stata anche considerata l'opzione qualora fosse inserito uno user inesistente.

addAntena.php-

Questa è la pagina che elabora i dati dopo aver compilato la pagina predefinita per aggiungere un'antenna nella pagina "Administrator.php". Una volta riempito ogni campo nella pagina administrator.php, questi dati sono prelevati tramite il metodo POST in questa suddetta pagina e, a seguire, con una semplice query INSERT, aggiungiamo ogni dato viene inserito in una nuova riga nella tabella antenne, creandone così una nuova.

```
addAntena.php
1 <?php
2 include "lidhjaDB.php";
3 $us=$_POST['username'];
4 $pwd=$_POST['pwd'];
5 $type=$_POST['type'];
6 $info=$_POST['info'];
7 // Check connection
8 if (!$conn) {
9     die("Connection failed: " . mysqli_connect_error());
10 }
11
12 $result1 = mysqli_query($conn, "SELECT MAX(id_antena) AS MAX FROM antena"); //sceglie max
13 $row1 = mysqli_fetch_assoc($result1);
14 $id= $row1["MAX"]+1; //max aggiunge 1
15 $sql = "INSERT INTO antena (id_antena,username,password,tipi,info) VALUES ($id,'$us', '$pwd','$type', '$info')"; //aggiunge le data prese dall inputs nella tabell
16 if (mysqli_query($conn, $sql)) { //vede se l'operazione e stato completato con successo antenna
17     echo "Antena was successfully aded"; //stampa questa
18     header("Refresh: 2; URL = Administrator.php"); //ti porta alla pagina adim
19 } else {
20     echo "Error: " . $sql . "<br>" . mysqli_error($conn); //stampa l'errore
21     header("Refresh: 2; URL = Administrator.php"); // ti porta alla pagina admin
22 }
23 mysqli_close($conn);
24 ?>
```

admin_func_shto.php-

Questa pagina serve per creare il modo di aggiungere le antenne ,per questo usiamo una modalità detta popup.In base creiamo un unico funzione che in se stesso non è nient' altro che il modo implementato per aggiungere questi utenti. Nella nostra pagina administrator.php noi includiamo questa pagina e poi richiamiamo la funzione che ha descritta precedentemente.

```
admin_func_shto.php
1 <?php
2 function shtoUser(){
3     ?>
4     <!-- Modal User -->
5     <div class="modal fade" id="myModal" role="dialog">
6     <div class="modal-dialog">
7     <!-- Modal content-->
8     <div class="modal-content">
9     <div class="modal-header">
10    <button type="button" class="close" data-dismiss="modal"><span></span></button>
11    <h4>Add Antena</h4>
12    </div>
13    <div class="modal-body">
14    <form id="shto" action="addAntena.php" method="post" class="form-horizontal" enctype="multipart/form-data" role="form">
15
16    <div class="form-group">
17    <label for="username"><span class="glyphicon glyphicon-user"></span>Username</label>
18    <input type="text" class="form-control" name="username" id="username" placeholder="username" autofocus required>
19    </div>
20
21    <div class="form-group">
22    <label for="password"><span class="glyphicon glyphicon-user"></span>Password</label>
23    <input type="password" class="form-control" name="pwd" id="pwd" placeholder="Password" autofocus required>
24    </div>
25
26    <div class="form-group">
27    <label for="type"><span class="glyphicon glyphicon-user"></span>Type</label>
28    <input type="text" class="form-control" name="type" id="type" placeholder="Type" autofocus required>
29    </div>
30
31    <div class="form-group">
32    <label for="info"><span class="glyphicon glyphicon-comment"></span>Info</label>
33    <textarea rows="4" cols="50" name="info" form="shto"></textarea>
34    </div>
35    <br>
36    <button type="submit" class="btn btn-block" Add Antena<span class="glyphicon glyphicon-ok"></span>
37    </button>
38    </form>
39    </div>
40    <div class="modal-footer">
41    <button type="submit" class="btn btn-danger btn-default pull-left" data-dismiss="modal"> <span class="glyphicon glyphicon-remove"></span> Cancel</button>
42    </div>
43    </div>
```

Administrator.php-

È l'interfaccia principale attraverso la quale possiamo visualizzare le informazioni degli utenti e delle antenne. Qui possiamo aggiungere altre antenne cliccando add_antenna nella parte sinistra della pagina. Dopo di che appare un form con alcune aree da compilare per la creazione della nuova antenna.

antennaTable.php-

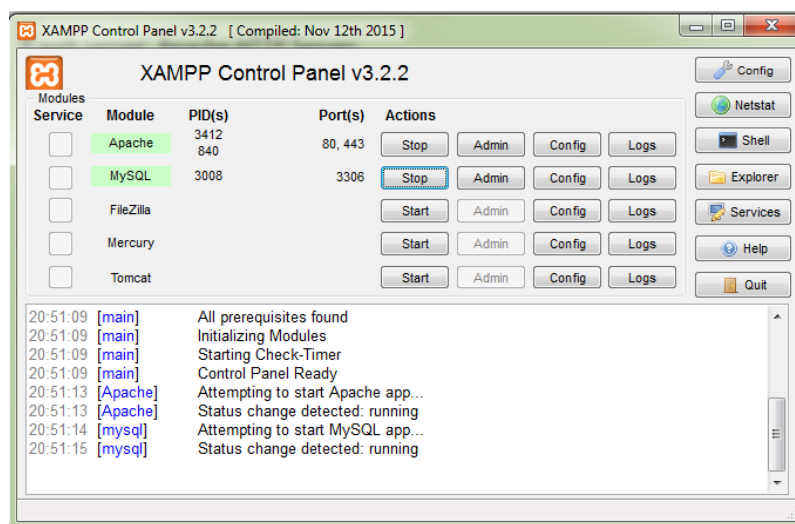
Questo file php viene utilizzato per visualizzare i dati della tabella antenne nella pagina di amministrazione. Inizialmente includiamo la pagina che ci permette di fare il collegamento con il database "lidhjadb.php" e dopo creiamo una query dove prendiamo tutte le colonne che siamo interessati a visualizzare, poi li prendiamo come un vettore e li visualizziamo riga dopo riga tramite la funzione "echo".

Funzionamento Pratico

Definizione XAMPP-XAMPP: è una piattaforma software multiplatforma e libera costituita da Apache HTTP Server, il database MariaDB e tutti gli strumenti necessari per utilizzare i linguaggi di programmazione PHP e Perl. Il nome è un acronimo dei software sopra citati (la X sta per x-platform, l'abbreviazione di cross-platform in lingua inglese ovvero multiplatforma). XAMPP è un software libero rilasciato attraverso GNU General Public License ed è caratterizzato da un approccio user friendly. Mediante XAMPP è possibile avere un application server capace di interpretare pagine web dinamiche. Il software è disponibile per sistemi unix-like (GNU/Linux, Oracle Solaris, MacOS, ...) e per Microsoft Windows. I componenti di base sono:

1. Il web server: Apache HTTP Server;
2. Il database management system (o database server): MariaDB e SQLite;
3. Il server FTP: ProFTPD;
4. Il mail server: Mercury Mail Transport System (solo per Microsoft Windows);
5. Come linguaggio di programmazione: Perl, PHP e/o Python.

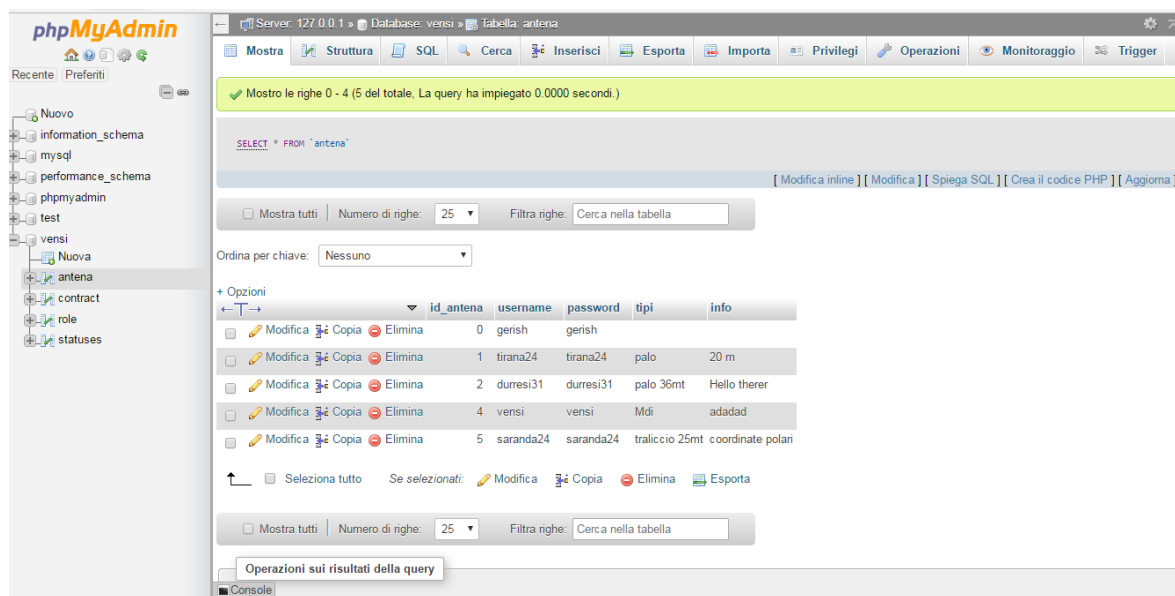
Quello che va usato in questo programma è l'Apache server e http e anche mySQL per testare la pagina e per fare i collegamenti con il database.



Definizione phpMyAdmin: è un'applicazione web scritta in PHP, distribuita con licenza GPL, che consente di amministrare un database MySQL o MariaDB tramite un qualsiasi browser. L'applicazione è indirizzata sia agli amministratori del database, sia agli utenti. Gestisce i permessi prelevandoli dal database. phpMyAdmin permette di creare un database da zero, creare le tabelle ed eseguire operazioni di ottimizzazione sulle stesse. Presenta un feedback sulla creazione delle tabelle per evitare eventuali errori. Sono previste delle funzionalità per l'inserimento dei dati (popolazione del database), per le query, per il backup dei dati, ecc.

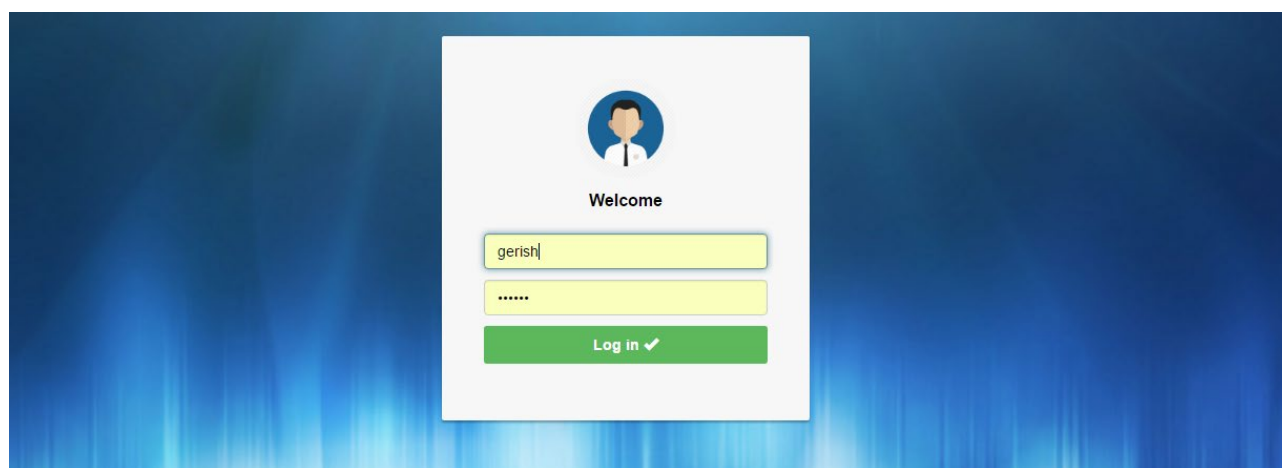
L'amministratore ha anche a disposizione un'interfaccia grafica per la gestione degli utenti: l'interfaccia permette l'inserimento di un nuovo utente, la modifica della relativa password e la gestione dei permessi che l'utente ha sul database. Requisiti necessari

1. Un server web (solitamente Apache HTTP Server);
2. Pacchetto MySQL o MariaDB server;
3. Pacchetto PHP;
4. Un browser.



Funzionamento pratico

Entriamo nella pagina WEB. Visualizziamo dunque una una tabella con 2 campi che vanno dove inserire username e password.

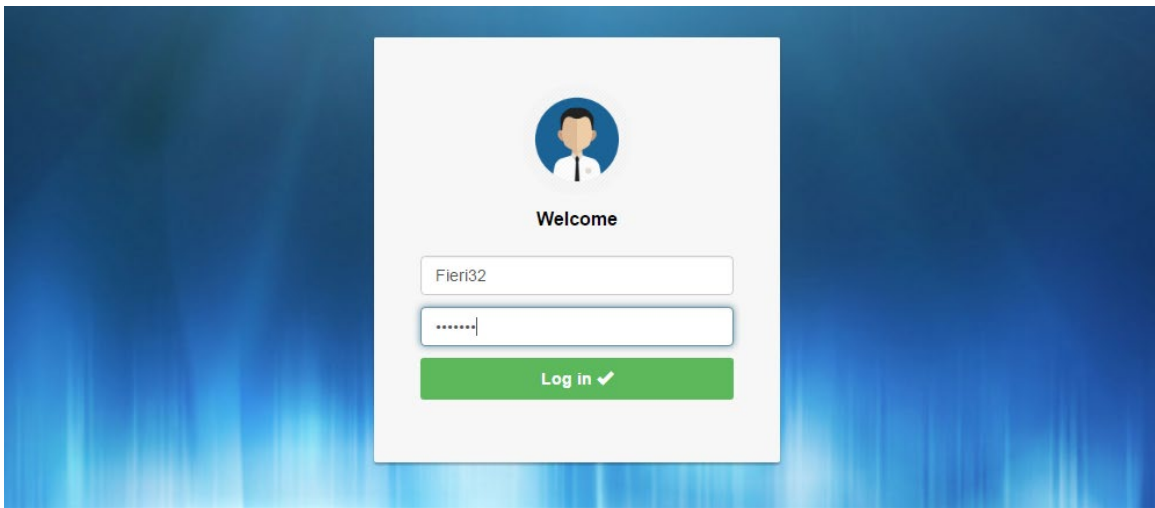


Come evidenziato precedentemente, abbiamo 2 tipi di user:

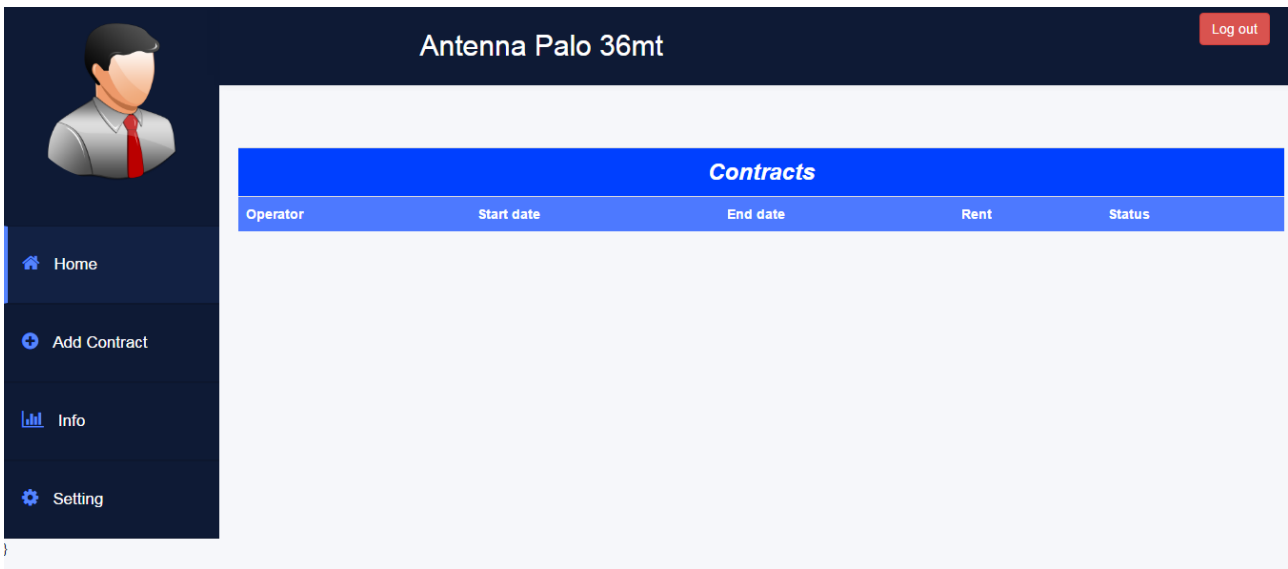
1. Amministratore
2. User semplice

L'amministratore prima deve aggiungere l'antenna che vale come user semplice.

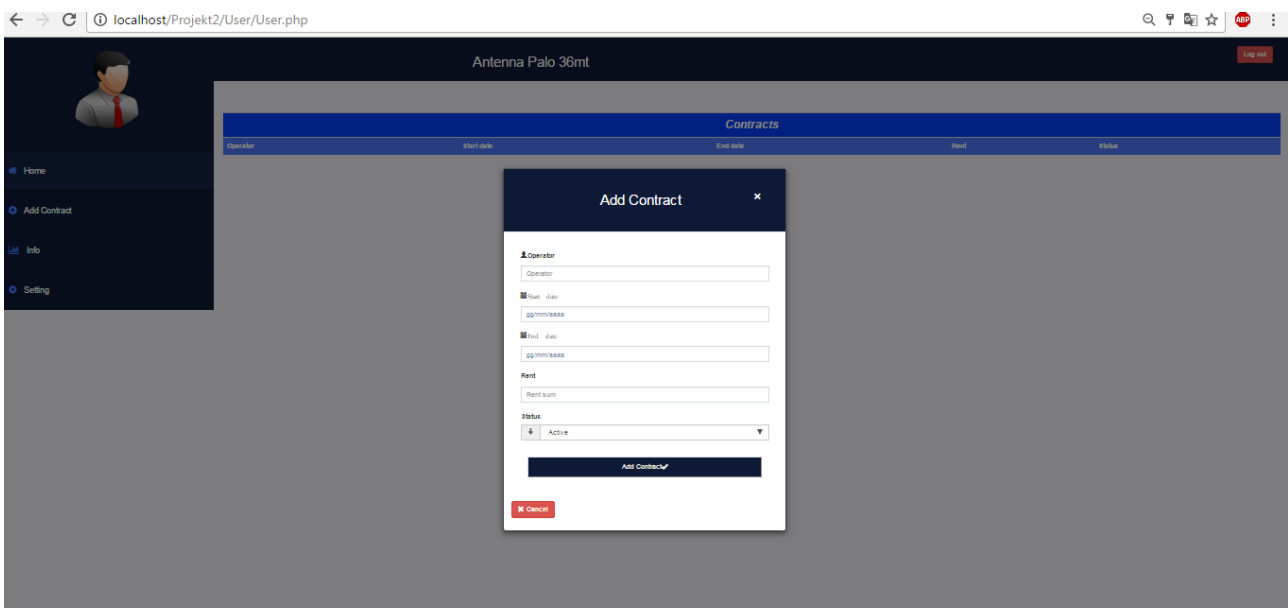
Dunque prendiamo in considerazione un user semplice ed entriamo e vediamo passo a passo che succede.



Facciamo il login. Una Volta fatto il login vediamo che ci si apre un'altra schermata.



Qui possiamo aggiungere i contratti che fanno parte di questa antenna. Premendo il bottone "Add Contract" attraverso quest'altra schermata:



Una volta aggiunti i contratti possiamo visualizzare un altro tipo di schermata con i contratti presenti, che possono essere Active or Disabled. Active lo si trova quando il contratto è ancora attivo e disabled quando non è più attivo.

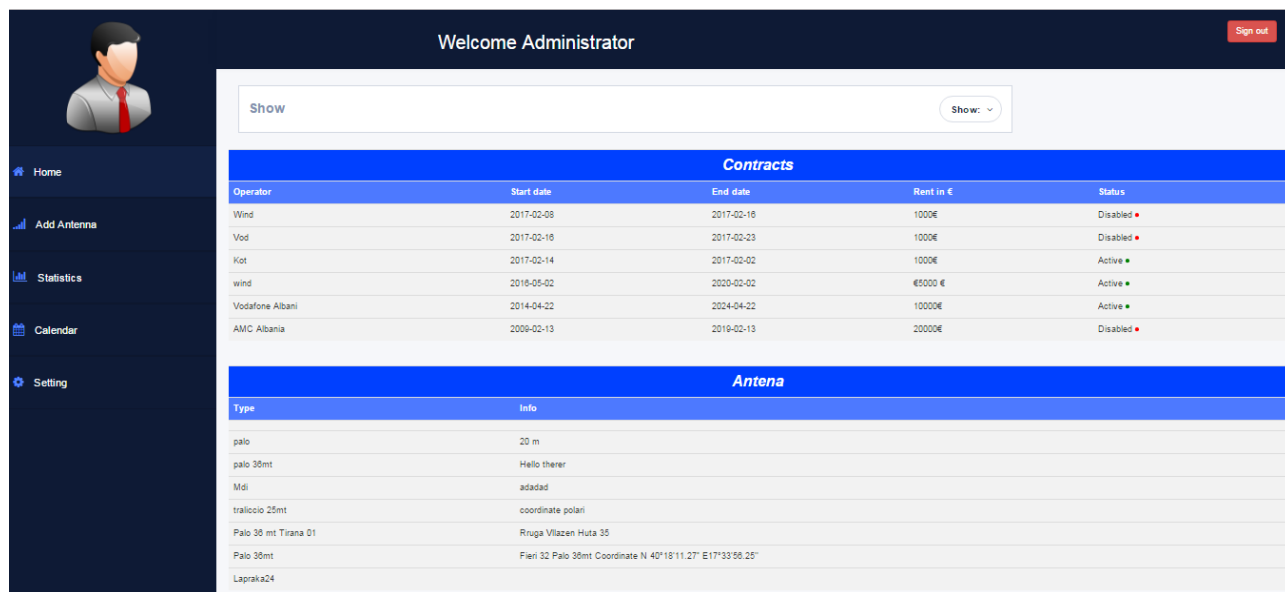


The screenshot shows a dashboard for 'Antenna Palo 36mt'. On the left is a dark sidebar with a user profile icon and menu items: Home, Add Contract, Info, and Setting. The main content area has a blue header 'Contracts' and a table with the following data:

Operator	Start date	End date	Rent	Status
Vodafone Albani	2014-04-22	2024-04-22	10000	Active ●
AMC Albania	2009-02-13	2019-02-13	20000	Disabled ●

Poi, da qui possiamo fare il log out, ritornando nella pagina iniziale.

Facendo invece il login come , ritroviamo una diversa schermata.



The screenshot shows a 'Welcome Administrator' dashboard. It features a 'Sign out' button in the top right corner and a search bar with a 'Show' dropdown menu. Below the search bar are two tables:

Contracts Table:

Operator	Start date	End date	Rent in €	Status
Wind	2017-02-08	2017-02-18	1000€	Disabled ●
Vod	2017-02-18	2017-02-23	1000€	Disabled ●
Kot	2017-02-14	2017-02-02	1000€	Active ●
wind	2018-05-02	2020-02-02	€5000	Active ●
Vodafone Albani	2014-04-22	2024-04-22	10000€	Active ●
AMC Albania	2009-02-13	2019-02-13	20000€	Disabled ●

Antena Table:

Type	Info
palo	20 m
palo 36mt	Hello there!
Mdi	adadad
traliccio 25mt	coordinate polari
Palo 36 mt Tirana 01	Rruga Vilazen Huta 35
Palo 36mt	Fieri 32 Palo 36mt Coordinate N 40°18'11.27" E17°33'58.25"
LapraKa24	

Attraverso questa pagina visualizziamo tutti i contratti presenti e tutte le antenne realizzate. Nella parte alta della pagina vediamo il comando Show.

Abbiamo 3 possibilità di visualizzazione: contratti e antenne, solo i contratti o solo le antenne.

Welcome Administrator

Show

Show: ▾
Contracts
Antenas
All

Contracts				
Operator	Start date	End date	Rent in €	Status
Wind	2017-02-08	2017-02-16	1000€	Disabled ●
Vod	2017-02-16	2017-02-23	1000€	Disabled ●
Kot	2017-02-14	2017-02-02	1000€	Active ●
wind	2016-05-02	2020-02-02	€5000 €	Active ●
Vodafone Albani	2014-04-22	2024-04-22	10000€	Active ●
AMC Albania	2009-02-13	2019-02-13	20000€	Disabled ●

Per ogni selezione poi cambia pure la visualizzazione.

Per i solo contratti abbiamo questa schermata:

Welcome Administrator Sign out

Show

Show: ▾

Contracts				
Operator	Start date	End date	Rent in €	Status
Wind	2017-02-08	2017-02-16	1000€	Disabled ●
Vod	2017-02-16	2017-02-23	1000€	Disabled ●
Kot	2017-02-14	2017-02-02	1000€	Active ●
wind	2016-05-02	2020-02-02	€5000 €	Active ●
Vodafone Albani	2014-04-22	2024-04-22	10000€	Active ●
AMC Albania	2009-02-13	2019-02-13	20000€	Disabled ●

Per solo le antenne:

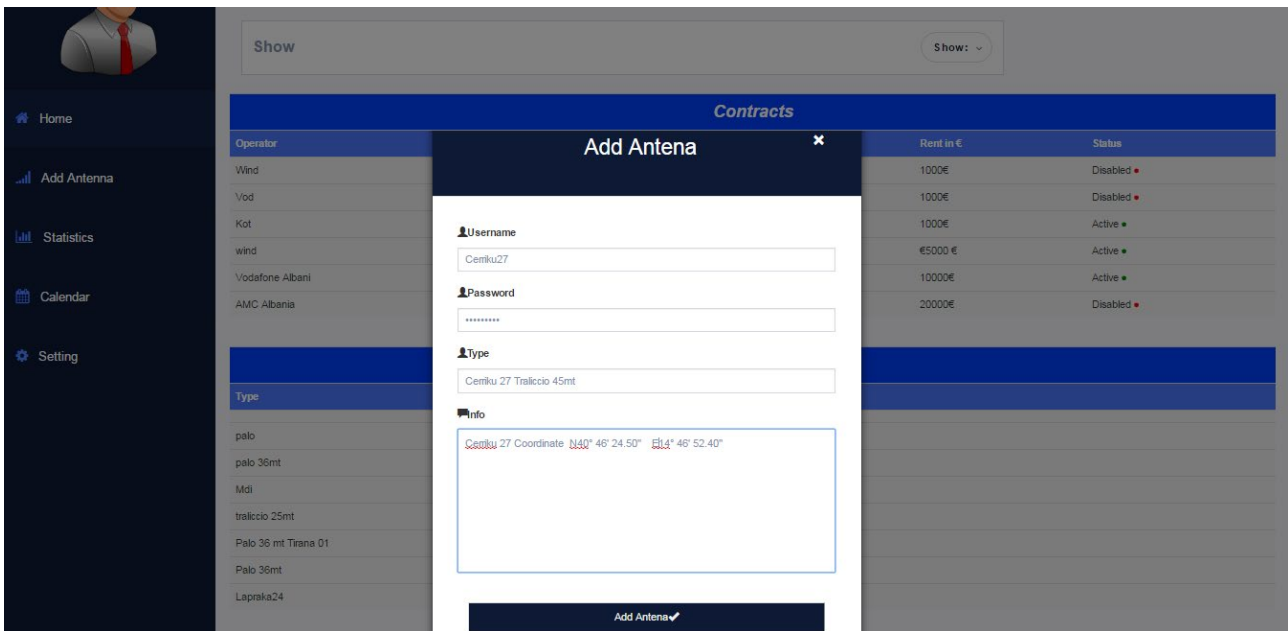
Welcome Administrator Sign out

Show

Show: ▾

Antena	
Type	Info
palo	20 m
palo 36mt	Hello therer
Mdi	edadad
traliccio 25mt	coordinate polari
Palo 36 mt Tirana 01	Rruga Vllazen Huta 35
Palo 36mt	Fieri 32 Palo 36mt Coordinate N 40°18'11.27" E17°33'56.25"
Lapraka24	

In modalità di amministratore, come già evidenziato, premendo sul tasto Add Antenna si può inserire una nuova antenna.



E vediamo che poi sarà presente nella lista delle antenne:

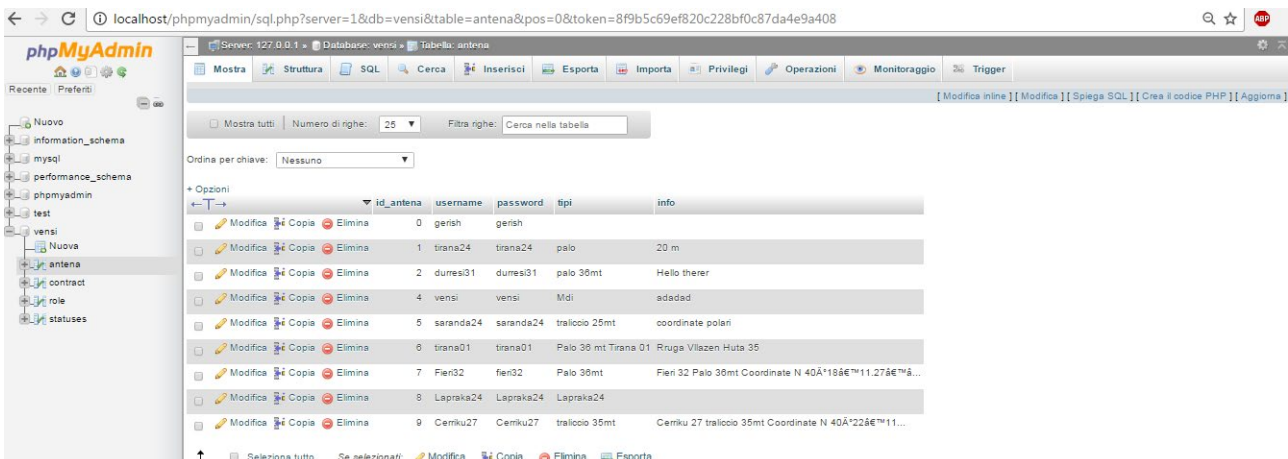


Per aggiungere i contratti relativi a questa antenna, bisogna seguire la stessa procedura che si segue entrando come user.

Le stesse cose si possono visualizzare anche nel database di phpMyAdmin.

Vediamo una breve descrizione del database e i collegamenti con la pagina web.

Per l'antenna:



Come si può notare, sono presenti gli stessi campi compilati per la creazione di nuovi user (username, la password assegnata ad ogni utente , il tipo antenna descrizione).

La pagina contract del database:

	id_contract	operator	start	end	rent	status	antenna_id
<input type="checkbox"/>	1	Wind	2017-02-08	2017-02-18	1000	1	1
<input type="checkbox"/>	2	Vod	2017-02-16	2017-02-23	1000	1	2
<input type="checkbox"/>	3	Kot	2017-02-14	2017-02-02	1000	0	1
<input type="checkbox"/>	4	wind	2018-05-02	2020-02-02	~5000	0	5
<input type="checkbox"/>	5	Vodafone Albani	2014-04-22	2024-04-22	10000	0	7
<input type="checkbox"/>	6	AMC Albania	2009-02-13	2019-02-13	20000	1	7

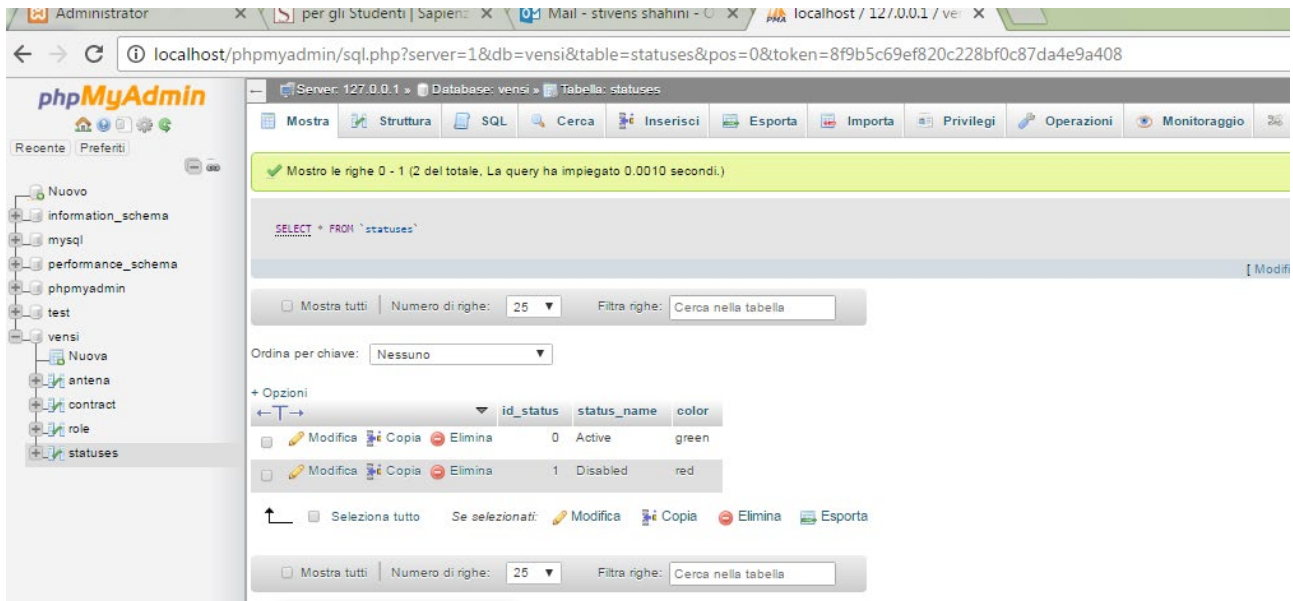
Pure qui riusciamo a veder i campi dei contratti descritti precedentemente. Si vede l'operatore, la data del inizio contratto, la data della fine e lo status.

Il ruolo:

	id_role	roli
<input type="checkbox"/>	1	user
<input type="checkbox"/>	2	admin

Possiamo vedere 2 id ruolo. Se sono 2, si ha a che fare con uno user di tipo admin, altrimenti si tratta di uno un user semplice.

Lo status:



Si possono avere 2 tipi di status:

1. Active
2. Disabled

Ognuno ha un colore rappresentativo.

Il colore verde è associato allo status Active e vuol dire che il contratto è attivo e dunque è in essere, mentre il colore rosso corrisponde al disabled e vuol dire che il contratto non è più attivo.

Creazione della tabella Antenna

```
CREATE TABLE `antenna` (
  `id_antenna` int(11) NOT NULL,
  `username` varchar(30) NOT NULL,
  `password` varchar(30) NOT NULL,
  `tipi` varchar(20) NOT NULL,
  `info` varchar(150) NOT NULL
)
```

Creazione della tabella contract

```
CREATE TABLE `contract` (
```

```
`id_contract` int(11) NOT NULL,  
`operator` varchar(15) NOT NULL,  
`start` date NOT NULL,  
`end` date NOT NULL,  
`rent` varchar(20) NOT NULL,  
`status` smallint(6) NOT NULL,  
`antenna_id` int(11) NOT NULL  
)
```

Esempi di aggiungere nella tabella contract

```
INSERT INTO `contract` (`id_contract`, `operator`, `start`, `end`, `rent`, `status`, `antenna_id`) VALUES  
(1, 'Wind', '2017-02-08', '2017-02-16', '1000', 0, 1),  
(2, 'Vod', '2017-02-16', '2017-02-23', '1000', 1, 2),  
(3, 'Kot', '2017-02-14', '2017-02-02', '1000', 0, 1);
```

creazione della tabella role, inserimento dei roli

```
CREATE TABLE `role` (  
  `id_role` smallint(6) NOT NULL,  
  `roli` varchar(10) NOT NULL  
)
```

```
INSERT INTO `role` (`id_role`, `roli`) VALUES  
(1, 'user'),  
(2, 'admin');
```

Creazione della tabella status e l'inserimento dei status

```
CREATE TABLE `statuses` (  
  `id_status` smallint(6) NOT NULL DEFAULT '0',  
  `status_name` varchar(15) NOT NULL,  
  `color` varchar(10) NOT NULL
```

)

```
INSERT INTO `statuses` (`id_status`, `status_name`, `color`) VALUES  
(0, 'Active', 'green'),  
(1, 'Disabled', 'red');
```

Creazione del collegamento fra la tabella contract, gli i status e le antenne mediante le chiavi esterne (Foreign KEY)

```
ALTER TABLE `contract`  
  ADD CONSTRAINT `contract_ibfk_1` FOREIGN KEY (`status`) REFERENCES `statuses` (`id_status`) ON  
  DELETE NO ACTION ON UPDATE NO ACTION,  
  ADD CONSTRAINT `contract_ibfk_2` FOREIGN KEY (`antenna_id`) REFERENCES `antenna` (`id_antenna`) ON  
  DELETE NO ACTION ON UPDATE NO ACTION;
```