

DESCRIZIONE C.M.S. MICROHARD E SUA INSTALLAZIONE

Il C.M.S. è stato sviluppato sullo STACK LAMP, quindi Linux come sistema operativo, Apache come web server, MariaDB come RDBMS e PHP come linguaggio lato back-end.

Per il front end invece è stato basato, su COREUI, che ha dato la possibilità di sviluppare in modo molto veloce, affinché si potesse creare una dashboard moderna per qualsiasi piattaforma, browser o dispositivo. Un kit UI completo che ti consente di creare rapidamente applicazioni reattive accattivanti, di alta qualità e ad alte prestazioni.

Nello specifico il back end è stato realizzato utilizzando il pattern MVC (Model View Controller) di seguito una breve sintesi del perché si è usato tale pattern.

Il pattern architetturale MVC (Model-View-Controller) è ampiamente utilizzato nella progettazione e sviluppo di applicazioni software e conviene per molte ragioni.

Innanzitutto, il pattern MVC separa l'applicazione in tre componenti distinti: il modello (Model), la vista (View) e il controllore (Controller). Questa separazione garantisce una maggiore modularità, manutenibilità e scalabilità dell'applicazione.

Il Model rappresenta i dati dell'applicazione e la logica di business. La View è responsabile della presentazione dei dati e dell'interfaccia utente. Il Controller si occupa della gestione degli eventi e della sincronizzazione tra il Model e la View.

In questo modo, il pattern MVC permette di creare una netta separazione tra la logica di business e la presentazione dei dati. Ciò significa che le modifiche alla logica di business non influenzeranno la View e viceversa. Inoltre, il Controller funge da mediatore tra il Model e la View, garantendo una comunicazione coerente e sincronizzata.

Altre ragioni per cui conviene utilizzare il pattern MVC includono:

Promuove la riusabilità del codice: grazie alla separazione dei componenti, è più facile riusare il codice tra diverse parti dell'applicazione.

Favorisce il testing: poiché il Model, la View e il Controller sono separati, è possibile testarli separatamente, aumentando la qualità complessiva del codice e riducendo il rischio di errori.

Sviluppo collaborativo: il pattern MVC permette a diversi sviluppatori di lavorare su parti diverse dell'applicazione in modo indipendente, senza interferire con il lavoro degli altri.

Migliora l'esperienza dell'utente: separando la logica di business dalla presentazione dei dati, l'utente può interagire con l'applicazione in modo più fluido e intuitivo.

In sintesi, il pattern MVC conviene perché permette di creare applicazioni modulari, manutenibili, scalabili e di alta qualità, favorendo lo sviluppo collaborativo e migliorando l'esperienza dell'utente.

Il C.M.S. gestisce diversi livelli di accesso ognuno con una autenticazione a 2 fattori implementato attraverso l'uso della APP di GOOGLE AUTHENTICATOR liberamente scaricabile dagli store ANDROID e APPLE.

Di seguito i livelli di accesso con i vari menù abilitati a seconda del profilo:

***** ACCESSO SUPER ADMIN *****

Utenti->Accessi creazione utenti per primo accesso

Esercenti->Anagrafiche vado ad associare l'utente creato prima aggiungendo altri campi della anagrafica, ad es. P.IVA indirizzo, città---

Catalogo Generale-> Gestione del catalogo generale con automatica produzione del JSON in caso di modifiche/inserimento

Catalogo Esercenti-> Gestione di tutti i cataloghi esercenti con possibilità di filtri in base a macchina esercente

Macchine Gestione-> Gestione e creazione di nuove macchine da associare ad un esercente

Gestione Media-> Possibilità di modifica/cancellazione dei MEDIA dei vari esercenti

Catalogo Carica catalogo generale-> Carico da file XML del catalogo genrale diviso per tipologie di macchine

Profilo-> Possibilità di modificarsi i dati del proprio profilo compresa la password

***** ACCESSO ESERCENTE *****

Catalogo->Inserisci Inserimento di un prodotto selezionandolo dal catalogo generale ed inserendolo con eventuali modifiche nel proprio

Catalogo->Modifica Possibilità di modifica/cancellazione di un prodotto a livello di catalogo macchina

Macchine vetrine - possibilità di inserire/modificare un prodotto in base alla configurazione della macchine

Macchine canali - possibilità di gestione di ogni singolo canale con possibilità di definire l'alias e lo stato

Macchine configurazioni - possibilità di aggiungere o modificare i dati di una macchina

Macchine stampe - possibilità di aggiungere o modificare i dati di stampa

Media gestione - gestione dei media

Banner gestione - gestione parametri banner/immagini

Rete gestione - gestione dei parametri di rete

Contabilità visualizza - visualizzazione dei dati contabili di una macchina con possibilità di inviare i dati via mail e/o di azzerarli

Tickets gestione - visualizzazione e/o modifica dello stato di un ticket con possibilità anche di ricerca ed esportazione

Log visualizza - visualizzazione dei log delle macchine con possibilità anche di ricerca ed esportazione

Recovery JSON - visualizzazione ed eventuale scarico dei JSON ricevuti dalle varie macchine con ricerche ed esportazione dei dati...

E' stato poi implementato un automatismo per la ricezione dei dati in formato JSON da parte delle Vending verso il C.M.S. come pure la produzione di JSON lato C.M.S. che vengono generati automaticamente al rispettivo inserimento, modifica o cancellazione di un qualsiasi dato.

Sono state quindi esposte delle **API** che, chiamate dalla singola Vending, possono avviare automaticamente la presa in carico di un qualsiasi JSON/Servizio lato macchina da inviare al C.M.S.

Ogni chiamata ad una qualsiasi API del C.M.S. viene preventivamente registrata in una tabella del C.M.S. indicando data/ora della avvenuta chiamata, l'utente/esercente che l'ha chiamata e la relativa macchina alla quale si riferisce quel determinato JSON.

Tutto ciò, oltre che dare la possibilità di visionare chi ha aggiornato cosa, permette tramite pagina dedicata di poter anche "ripristinare" un determinato JSON presente sul C.M.S. per poi caricarlo manualmente sulla Vending.

Tutto il C.M.S. è stato sviluppato senza alcuna dipendenza da framework o librerie proprietarie particolari, in questo modo sarà più facile anche da, replicare/installare su qualsiasi server, che abbia configurato lo stack LAMP.

INSTALLAZIONE CMS.

Per la corretta installazione e configurazione del C.M.S. bisogna implementare lo stack LAMP (Linux Apache MySql/MariaDB Php) con le seguenti versioni:

Php = 7.4

MariaDB = 10.x

Linux = qualsiasi distribuzione

Apache = qualsiasi distribuzione

Creare un database e poi importare (as es. tramite phpMyadmin) lo script SQL presente nella Root del pacchetto con il nome: **script_installazione_db.sql**

dopo aver installato il DB si potrà accedere con i seguenti dati (modificabili dopo il login dal menù profilo in alto a destra...)

Accesso Super Admin (permette la creazione di altri admin)

email = superadmin@inidirzzo.it

password = 1q2w3e4r

—

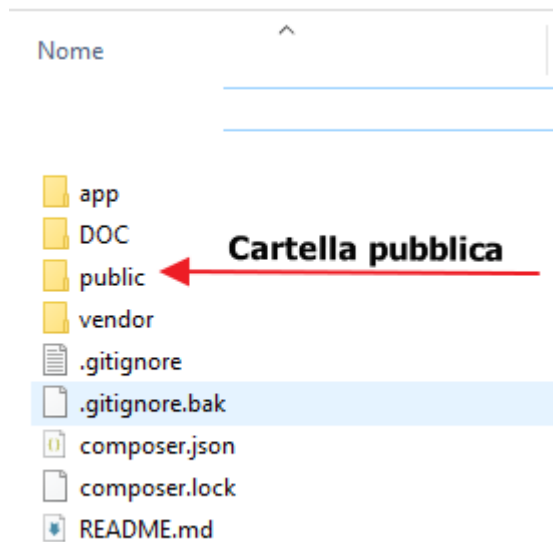
Accesso Admin (permette la creazione di esercenti)

email = dmin@inidirzzo.it
password = 1q2w3e4r

per il LOGIN bisogna scaricare sul proprio cellulare l'APP **GOOGLE AUTHENTICATOR**.

Effettuare il trasferimento di tutti i file del pacchetto C.M.S. via FTP sul server sul quale si vuole installare il C.M.S.

la struttura delle cartelle del C.M.S. è la seguente:



quella indicata è la cartella che deve essere configurata sul server per accesso al C.M.S. quindi questa sarà la documentRoot del server o del VirtualHost nel caso siano configurati più host su Apache.

Una volta effettuato il trasferimento di tutto il pacchetto C.M.S. bisognerà modificare i parametri di connessione ad Database in **due files** come di seguito evidenziato:

nel file **app/class/class_connection.php**

```

<?php
/**
 * CMS PER CLIENTE MICROHARD
 *
 * CMS per la gestione lato Web delle Vending Machine
 *
 * @package      CMS MICROHARD
 * @subpackage   Some Subpackage
 * @category     CMS per MICROHARD
 * @author       XIMPLIA
 */

// Define database configuration

define("DB_HOST", "[redacted]");
define("DB_USER", "[redacted]");
define("DB_PASS", "[redacted]");
define("DB_NAME", "[redacted]");

```

inserire:

DB_HOST = IP DEL SERVER

DB_USER = Utente del database (con tutti i privilegi)

DB_PASS = Password accesso al DB

DB_NAME = Nome Database

E poi nel file **app/xcrud/xcrud_config.php**

inserire:

```

app > xcrud > xcrud_config.php > Xcrud_config
1  <?php /**
2  *
3  *  N U O V A - V E R S I O N E 1.7
4  *
5  * */
6  class Xcrud_config
7  {
8
9
10     public static $dbname = '[redacted]'; // Your database name
11     public static $dbuser = '[redacted]'; // Your database username
12     public static $dbpass = '[redacted]'; // // Your database password
13     public static $dbhost = '[redacted]'; // Your database host, 'localhost' is default.
14     //public static $dbport = '3307'; // Your database host, 'localhost' is default.
15

```

\$dbhost = IP DEL SERVER

\$dbuser = Utente del database (con tutti i privilegi)

\$dbpass = Password accesso al DB

\$dbname = Nome Database

La documentazione delle classi e dei loro metodi è raggiungibile dalla URL del C.M.S. seguito dall'indicazione della cartella DOC:

esempio: www.indirizzocms.it/DOC/