

INTRODUZIONE

Il nostro progetto nasce da una domanda semplice: come possiamo trasformare una nuova infrastruttura ferroviaria in una vera opportunità per il territorio?

Siamo degli studenti dell'istituto tecnico economico A. Lombardi di Airola, un paese della provincia di Benevento ubicato nel cuore della Valle Caudina, ricca di storia, cultura e tradizioni, ma che spesso resta fuori dai principali circuiti turistici per una difficoltà di mobilità locale.

Basta pensare che a pochi chilometri da Airola si consumò la famosissima battaglia delle Forche Caudine, dove l'esercito romano venne sconfitto dai fieri guerrieri sanniti, oppure ai borghi storici di Sant'Agata De Goti scenari di diversi film tra cui ultimo uno dell'attore Siani, oppure al castello e alla torre di Montesarchio, che ne loro museo custodiscono tra le tante meraviglie dell'antichità, il vaso di Assteas, raffigurante il ratto di Europa, da molti definito il vaso più bello del mondo.



Dopo anni di isolamento finalmente a gennaio 2027 sarà riaperta la linea ferroviaria che collega Napoli a Benevento, passando proprio attraverso la nostra Valle Caudina!

💡 L'IDEA

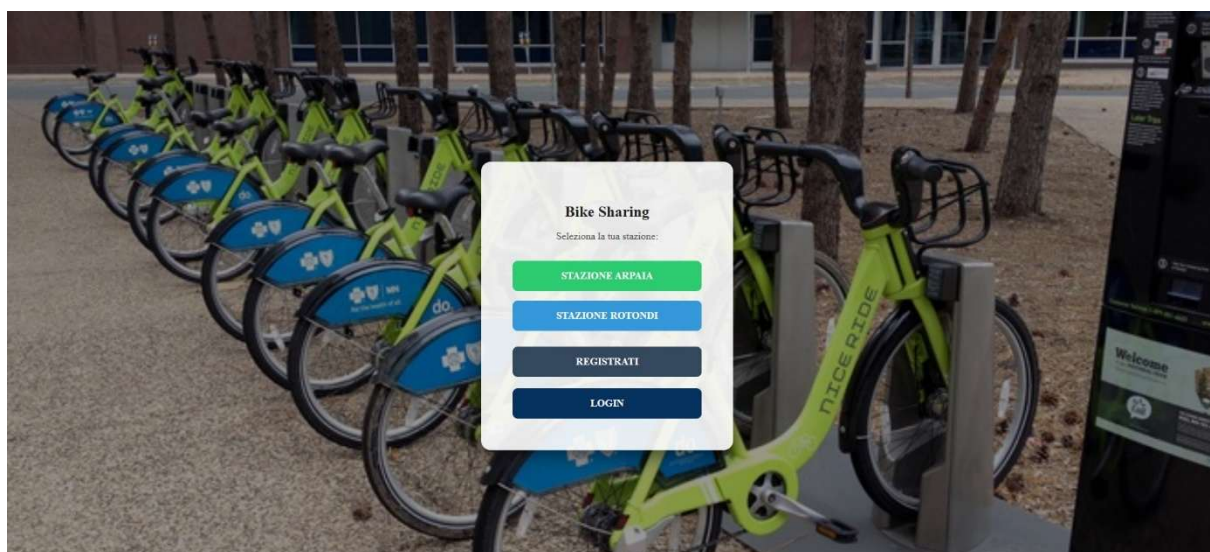
Abbiamo progettato un sistema di bike sharing intelligente basato su Arduino, da installare nelle stazioni della futura linea ferroviaria Benevento–Napoli.

L'obiettivo è permettere a turisti e cittadini che utilizzano il treno, di muoversi facilmente tra i paesi, in modo sostenibile e accessibile, in modo da favorire non solo la visita di tutte le nostre meraviglie, ma anche l'economia del territorio dando un nuovo impulso al turismo.

⚙️ COME FUNZIONA



La caudina bike sharing ha pensato di creare un apposito sito internet che tiene traccia delle bici disponibili, con la seguente schermata iniziale di benvenuto.



Quando il turista proveniente dai paesi limitrofi o dalle città di Benevento e Napoli si trova in treno, può consultare il sito internet su cui sono visibili le varie stazioni dove il treno effettua le fermate e la quantità di bici disponibili in real time.

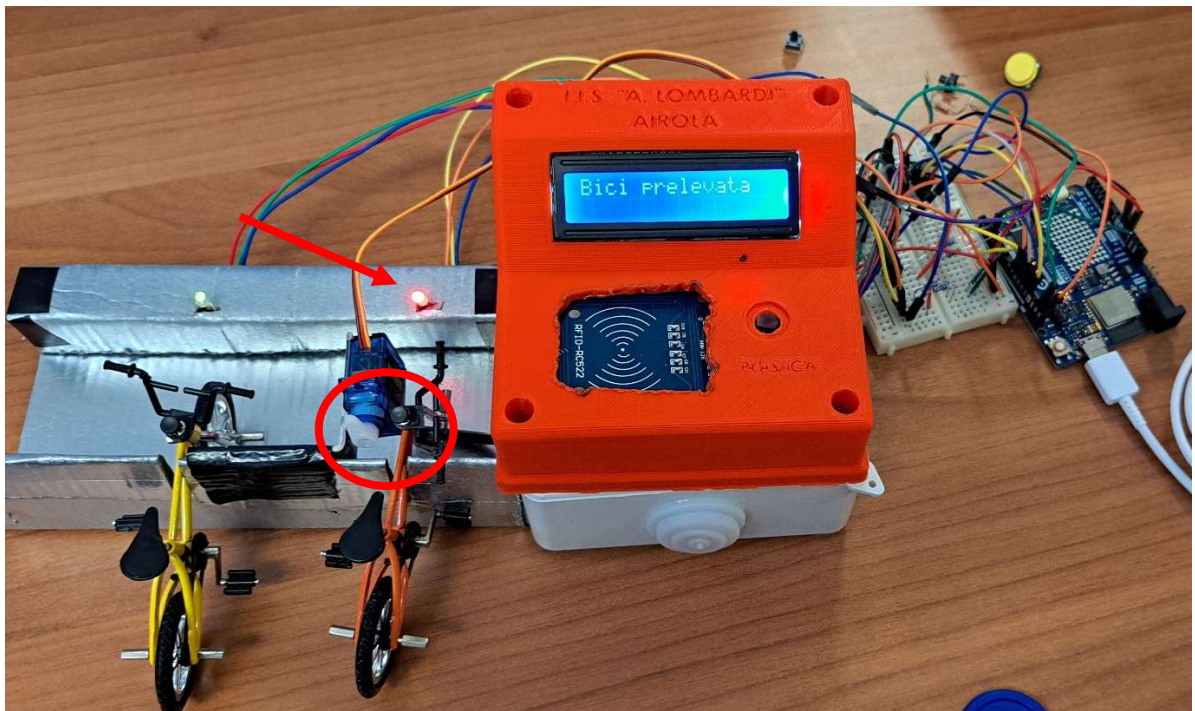
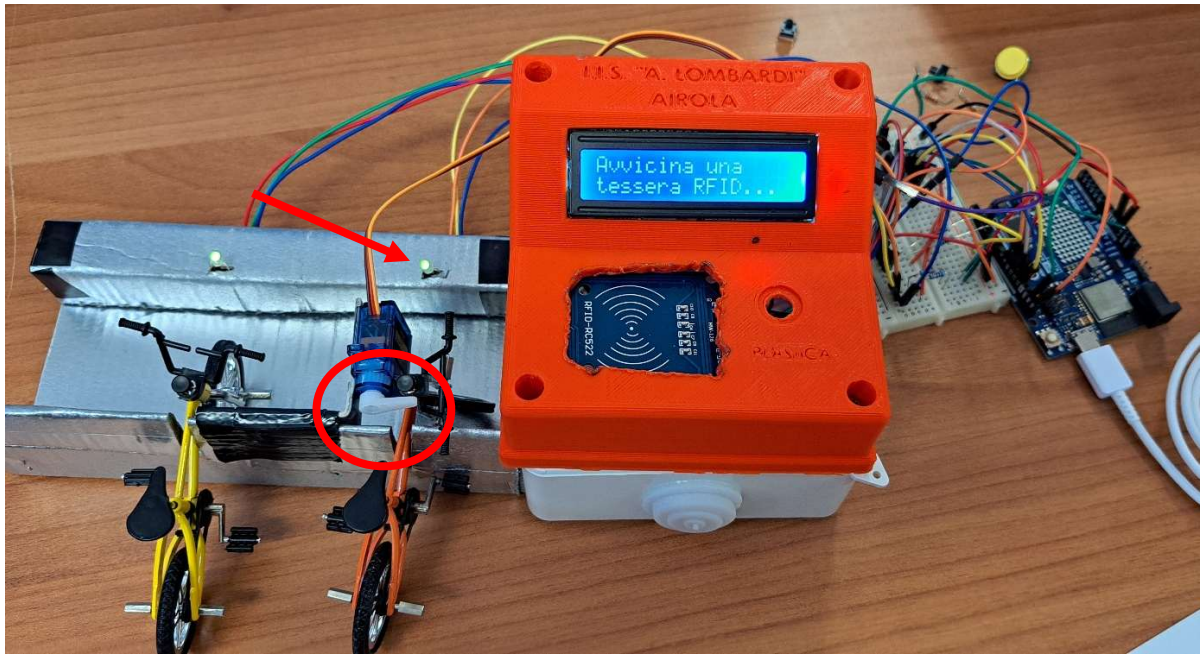


La dicitura “stato occupato” indica che nello stallo corrispondente è presente una bici che può essere prelevata. Nel caso dell’immagine sopra vediamo che tutte le cinque biciclette della stazione di Arpaia sono disponibili.

All’atto della registrazione (che immaginiamo avvenga agli sportelli della stazione) i dati dell’utente vengono inseriti in un database e gli viene rilasciata una carta RFID personale contenete dei crediti a pagamento. Ogni credito avrà un costo di 1 centesimo e consente di noleggiare la bici per un minuto.

uid_tessera	nome	cognome	DataNascita	numeroTel	email	città	indirizzo	crediti
01BE271C	Raffaele	Perrotta	2010-02-09	3916471467	raffaele.perrotta.s@iislombardi.edu.it	Airola	Via Rossi N 4	994
6342E613	Vincenzo	Giaquinto	2010-07-03	3289764162	vincenzo.giaquinto.s@iislombardi.edu.it	Bucciano	Via Roma N 31	852
93A9C395	Fabio	Simeone	2010-04-24	3459087213	fabio.simeone.s@iislombardi.edu.it	Arpaia	Via Augusto N 20	989
C3AF0328	Giovanni	Laudato	2010-12-16	3681324567	giovanni.laudato.s@iislombardi.edu.it	Moiano	Via Annunziata N 8	813

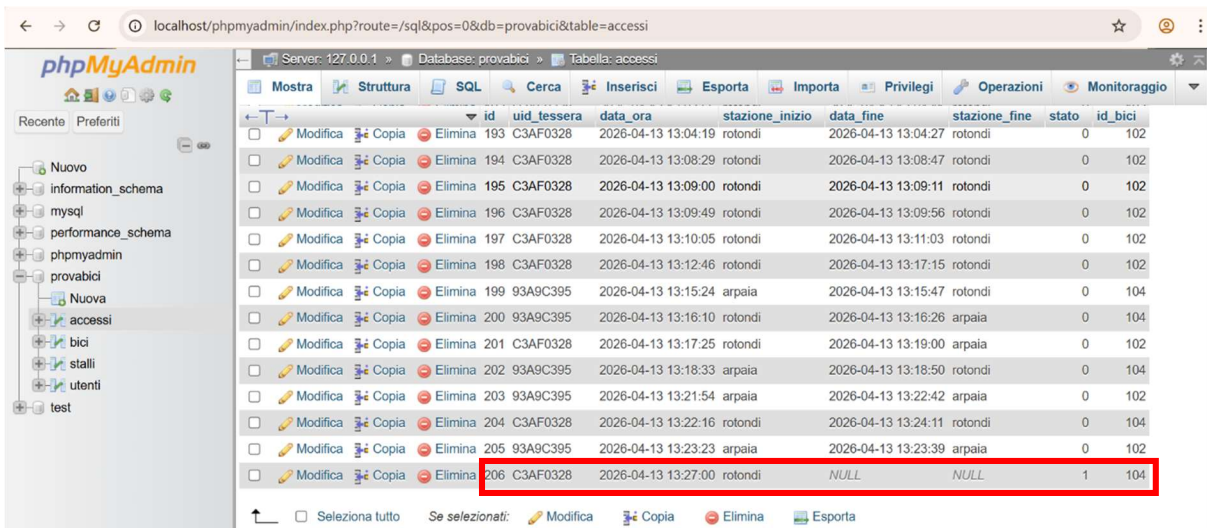
Avvicinando la carta al lettore situato in prossimità dello stallo, Arduino riconosce l’utente e sblocca automaticamente la prima bici disponibile, ruotando il servomotore di 90° e segnalando con la luce del led rossa che la bici è stata prelevata. Allo stesso tempo il sistema invia al database che gestisce i dati, tramite una connessione ad internet, l’id della tessera letta e la stazione in cui si trova.



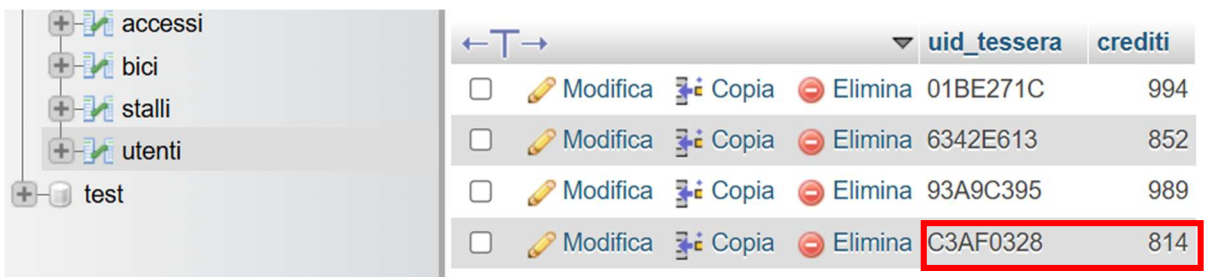
A questo punto il sito internet segnala immediatamente che una bici è stata prelevata e che lo stallo che la ospitava è libero di accogliere una nuova bici da posare. (figura sotto)



Allo stesso tempo il database tiene traccia del fatto che la bici 1 è attualmente impegnata, aggiungendo una nuova riga alla tabella “accessi” come si può vedere dalla figura sotto.

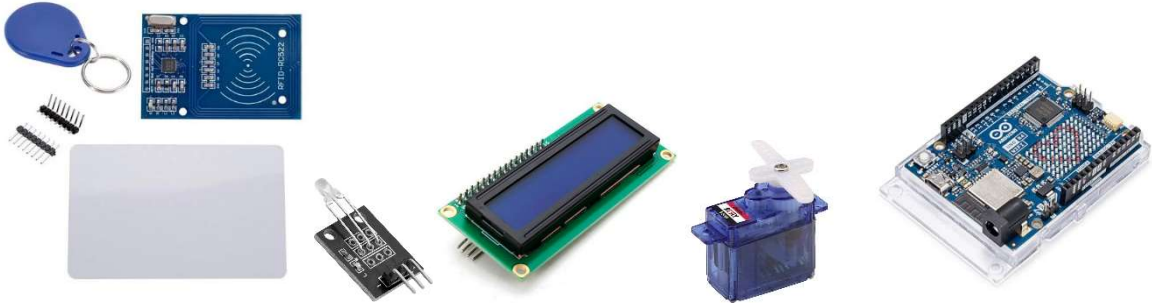


Alla riconsegna presso lo stallo di destinazione, il sistema blocca la bici e invia i dati relativi alla tessera e alla stazione finale al database, che gestisce così i crediti in base alla durata del noleggio nella tabella “utenti”.

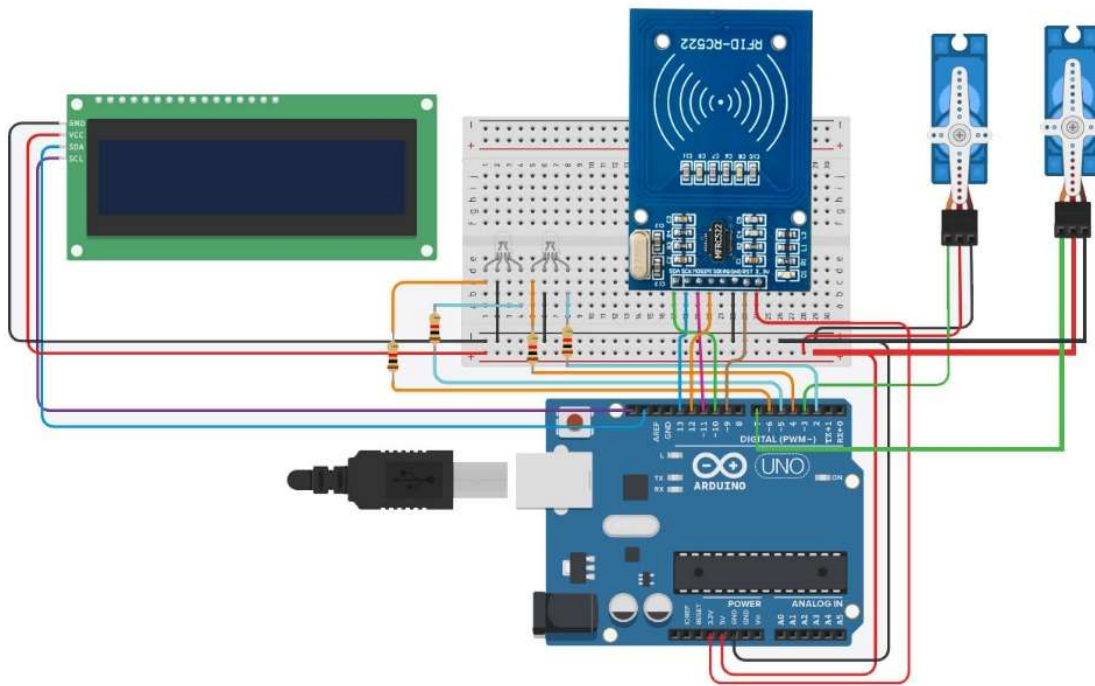


I componenti hardware utilizzati per il nostro Sistema sono i seguenti:

- modulo RFID
- led bicolore (green/red)
- Arduino R4 Wifi (che permette il collegamento ad internet)
- Schermino LCD I2C (16 x 2)
- Breadboard
- Restistenze
- Servomotori



Lo schema di collegamento è il seguente:



Abbiamo utilizzato ove possibile, componenti provenienti da altri progetti e convertiti ad hoc per il nostro caso, come ad esempio la colonnina per alloggiare il lettore RFID e il monitor LCD che originariamente era una pulsantiera da noi realizzata usando la stampante 3D.



Al fine di ottenere questi risultati il database è stato progettato per garantire:

- Tracciamento completo dei noleggi
- Associazione sicura utente ↔ tessera RFID
- Monitoraggio biciclette e stalli
- Gestione dei crediti

Questo permette ad Arduino di interagire con il sistema in tempo reale.

Abbiamo modellato il sistema attraverso le entità Utente, Accessi, Bici, e Stalli.

Le relazioni principali permettono di tracciare l'utilizzo delle biciclette, associando ogni noleggio a un utente, una bici e agli stalli di partenza e arrivo.

IMPATTO

Questo progetto non è solo tecnologico, ma sociale e ambientale.

- Promuove **mobilità sostenibile**
- Riduce traffico e inquinamento
- Incentiva il turismo lento
- Valorizza i borghi della Valle Caudina

Un turista che arriva in treno non trova solo una stazione, ma un sistema che lo accompagna alla scoperta del territorio.

VISIONE FUTURA

Attualmente abbiamo previsto di posizionare gli stalli nelle due stazioni ferroviarie più vicine a noi, ovvero quella di Arpaia-Airola-Sant'Agata e quella di Rotondi, ma il sistema è scalabile: può essere esteso ad altri comuni, integrato con app mobile e collegato ai servizi di trasporto pubblico.

Nel caso di diversi tipi di bici (elettrica o classica) andrà gestita la tariffazione differente.

Inoltre, per rendere il servizio più confortevole anche di notte, lo stesso arduino può comandare l'accensione delle luci per illuminare gli stalli usando un semplice fotoresistore ed un relè. (non è stato ancora inserito al momento per mancanza di piedini liberi sul nostro arduino r4 ma stiamo cercando di procurarci un arduino mega).

Il nostro obiettivo è trasformare questo progetto in un modello replicabile per altre aree interne d'Italia che presentino le nostre stesse criticità nei trasporti.

Con questo progetto dimostriamo che la tecnologia, anche a basso costo come Arduino, può diventare uno strumento concreto di sviluppo territoriale.

Non stiamo solo costruendo un sistema di noleggio biciclette, ma un ponte tra innovazione, sostenibilità e valorizzazione del territorio.

Link al video del progetto

<https://drive.google.com/file/d/1Mt9d0nfQmSC11boQ1iAfnzM045DwWRFt/view?ts=69df3ffb>

CODICE UTILIZZATO

I codici più importanti sono:

HOME PAGE:

```
<!DOCTYPE html>
<html lang="it">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bike Sharing - Homepage</title>
  <style>
    * {
      box-sizing: border-box;
    }

    body, html {
      height: 100%;
      margin: 0;
      padding: 0;
    }

    .bg-image {
      /* Sostituisci 'sfondo.jpg' con il nome esatto della tua foto */
      background-image: url('bike.jpg');

      /* Imposta l'altezza a tutto schermo */
      height: 100%;

      /* Centra l'immagine e la scala per coprire tutto lo spazio senza distorcerla */
      background-position: center;
      background-repeat: no-repeat;
      background-size: cover;

      /* Rende lo sfondo fisso durante lo scroll (effetto professionale) */
      background-attachment: fixed;

      /* Visualizzazione flessibile per centrare il contenuto sopra */
      display: flex;
      justify-content: center;
      align-items: center;
    }

    /* Overlay per scurire leggermente la foto e rendere il testo leggibile */
    .overlay {
      background-color: rgba(0, 0, 0, 0.4);
      position: absolute;
      top: 0;
      left: 0;
      width: 100%;
      height: 100%;
      z-index: 1;
    }
  </style>
</head>
<body>
  <div class="bg-image">
    <div class="overlay">
      <h1>Bike Sharing</h1>
    </div>
  </div>
</body>
</html>
```

```

.container {
  position: relative;
  z-index: 2; /* Per stare sopra l'overlay */
  text-align: center;
  background: # rgba(255, 255, 255, 0.9);
  padding: 50px;
  border-radius: 15px;
  box-shadow: 0 10px 30px # rgba(0,0,0,0.5);
  width: 90%;
  max-width: 400px;
}

h1 { margin-bottom: 10px; color: # #222; font-size: 24px; }
p { color: # #555; margin-bottom: 30px; }

.btn-container {
  display: flex;
  flex-direction: column;
  gap: 15px;
}

.btn {
  text-decoration: none;
  padding: 15px;
  border-radius: 8px;
  color: white;
  font-weight: bold;
  text-transform: uppercase;
  transition: 0.3s;
  text-align: center;
}

.arpaia { background-color: # #2ecc71; }
.rotondi { background-color: # #3498db; }

.btn:hover {
  transform: scale(1.03);
  filter: brightness(1.1);
}
</style>
</head>
<body>

<div class="bg-image">
  <div class="overlay"></div>

  <div class="container">
    <h1>Bike Sharing</h1>
    <p>Seleziona la tua stazione:</p>

    <div class="btn-container">
      <a href="lettore_bici.php" class="btn arpaia">Stazione Arpaia</a>
      <a href="stazionerotondi.php" class="btn rotondi">Stazione Rotondi</a>
    </div>
  </div>
</div>

</body>
</html>

```

STAZIONE ARPAIA:

```
<?php
// 1. CONFIGURAZIONE E CONNESSIONE
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "provabici";

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

if ($conn->connect_error) {
    die("Connessione fallita: " . $conn->connect_error);
}

// 2. LOGICA DI GESTIONE (PRELIEVO/CONSEGNA)
$messaggio_operazione = "";
$arduino_msg = ""; // Messaggio pulito per Arduino
$uid = $_GET['uid'] ?? null;
$stazione_attuale = $_GET['stazioni'] ?? "arpaia";

if($uid) {
    // Trova utente e crediti
    $res_u = $conn->query("SELECT crediti FROM utenti WHERE uid_tessera = '$uid'");
    if ($res_u->num_rows == 0) {
        $conn->query("INSERT INTO utenti (uid_tessera, crediti) VALUES ('$uid', 200)");
        $crediti = 200;
    } else {
        $crediti = $res_u->fetch_assoc()['crediti'];
    }

    // Controlla se l'utente ha già una bici (CHECK-OUT)
    $check_noleggio = $conn->query("SELECT id, id_bici, data_ora FROM accessi WHERE uid_tessera = '$uid' AND stato = 1 LIMIT 1");

    if ($check_noleggio->num_rows > 0) {
        // --- USCITA: POSA LA BICI ---
        $noleggio = $check_noleggio->fetch_assoc();
        $id_acc = $noleggio['id'];
        $bici_da_posare = $noleggio['id_bici'];

        $res_stallo = $conn->query("SELECT numero_stallo FROM stalli WHERE stazione = '$stazione_attuale' AND stato = 0 LIMIT 1");

        if ($res_stallo->num_rows > 0) {
            $num_st = $res_stallo->fetch_assoc()['numero_stallo'];

            $conn->query("UPDATE utenti SET crediti = crediti - 1 WHERE uid_tessera = '$uid'");
        }
    }
}
```

```

        $conn->query("UPDATE accessi SET data_fine = NOW(), stazione_fine = '$stazione_attuale',
stato = 0 WHERE id = $id_acc");
        $conn->query("UPDATE stalli SET stato = 1, id_bici = $bici_da_posare WHERE stazione =
'$stazione_attuale' AND numero_stallo = $num_st");

        $arduino_msg = "posata stallo$num_st";
        $messaggio_operazione = "<div class='alert successo'>USCITA: Bici posata allo stallo
$num_st.</div>";
    } else {
        $arduino_msg = "ERRORE: Piena";
        $messaggio_operazione = "<div class='alert errore'>ERRORE: Stazione piena!</div>";
    }

} else {
    // --- INGRESSO: PRELEVA BICI ---
    if ($crediti > 0) {
        $res_bici = $conn->query("SELECT numero_stallo, id_bici FROM stalli WHERE stazione =
'$stazione_attuale' AND stato = 1 LIMIT 1");

        if ($res_bici->num_rows > 0) {
            $info = $res_bici->fetch_assoc();
            $id_b = $info['id_bici'];
            $num_st = $info['numero_stallo'];

            $conn->query("UPDATE stalli SET stato = 0, id_bici = NULL WHERE stazione =
'$stazione_attuale' AND numero_stallo = $num_st");
            $conn->query("INSERT INTO accessi (uid_tessera, stazione_inizio, id_bici, data_ora,
stato) VALUES ('$uid', '$stazione_attuale', $id_b, NOW(), 1)");

            $arduino_msg = "Prelevata stallo$num_st";
            $messaggio_operazione = "<div class='alert successo'>INGRESSO: Prelevata dallo stallo
$num_st.</div>";
        } else {
            $arduino_msg = "ERRORE: Vuota";
            $messaggio_operazione = "<div class='alert errore'>ERRORE: Nessuna bici
disponibile.</div>";
        }
    } else {
        $arduino_msg = "ERRORE: Crediti";
        $messaggio_operazione = "<div class='alert errore'>ERRORE: Crediti insufficienti.</div>";
    }
}
}

// 3. INVIO RISPOSTA AD ARDUINO E USCITA (Spostato qui per funzionare correttamente)
if (isset($_GET['arduino'])){
    echo $arduino_msg;
    $conn->close();
}

```

```

    exit;
}

// 4. FUNZIONE PER GENERARE LA TABELLA
function generaTabella($conn, $nomeStazione) {
    $sql = "SELECT numero_stallo, stato FROM stalli WHERE stazione = '$nomeStazione' ORDER BY
numero_stallo ASC";
    $result = $conn->query($sql);
    echo "<h3>Stazione di " . ucfirst($nomeStazione) . "</h3>";
    echo "<table><thead><tr><th>Stallo</th><th>Stato</th></tr></thead><tbody>";
    if ($result && $result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            $testo = ($row["stato"] == 0) ? "Libero" : "Occupato";
            $classe = ($row["stato"] == 0) ? "libero" : "occupato";
            echo "<tr><td>N. " . $row["numero_stallo"] . "</td><td class='$classe'$testo</td></tr>";
        }
    } else { echo "<tr><td colspan='2'>Nessun dato trovato</td></tr>"; }
    echo "</tbody></table>";
}
?>

<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <title>Monitoraggio - <?php echo ucfirst($stazione_attuale); ?></title>
    <style>
        body, html { height: 100%; margin: 0; font-family: 'Segoe UI', sans-serif; }
        .bg-container { background-image: url('arpaia2.jpg'); background-size: cover; background-
position: center; min-height: 100vh; display: flex; flex-direction: column; align-items: center;
justify-content: center; position: relative; }
        .overlay { position: absolute; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,
0, 0, 0.5); z-index: 1; }
        .content-box { position: relative; z-index: 2; background: rgba(255, 255, 255, 0.95); padding:
30px; border-radius: 20px; text-align: center; width: 90%; max-width: 450px; }
        h1 { color: white; position: relative; z-index: 2; text-shadow: 2px 2px 4px rgba(0,0,0,0.5); }
        table { border-collapse: collapse; width: 100%; background: white; border-radius: 10px;
overflow: hidden; margin-bottom: 20px; }
        th { background: #37474f; color: white; padding: 12px; }
        td { padding: 12px; border-bottom: 1px solid #eee; }
        .libero { color: #27ae60; font-weight: bold; background: #e8f5e9; }
        .occupato { color: #c0392b; font-weight: bold; background: #ffebee; }
        .alert { padding: 15px; margin-bottom: 20px; border-radius: 8px; font-weight: bold; }
        .successo { background: #d4edda; color: #155724; border: 1px solid #c3e6cb; }
        .errore { background: #f8d7da; color: #721c24; border: 1px solid #f5c6cb; }
    </style>
</head>
<body>

```

```

<div class="bg-container">
  <div class="overlay"></div>
  <h1>Monitoraggio Sistema Bici</h1>
  <div class="content-box">
    <?php echo $messaggio_operazione; ?>
    <div class="sezione-stazione"><?php generaTabella($conn, $stazione_attuale); ?></div>
    <br><a href="Homepagebici.php" style="text-decoration:none; color:#34495e;"><- Torna alla
Home</a>
  </div>
</div>
</body>
</html>
<?php $conn->close(); ?>

```

CODICE ARDUINO ARPAIA:

```

#include <WiFi3.h>
#include <WiFiUdp.h>
#include <ArduinoHttpClient.h>
#include <Arduino_ConnectionHandler.h>
#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal_PCF8574.h>
#define SS_PIN 10
#define RST_PIN 9
#include <Servo.h>

MFRC522 rfid(SS_PIN, RST_PIN);

char ssid[] = "Redmi Note 13 V";
char pass[] = "12345678";
WiFiConnectionHandler ArduinoIoTPreferredConnection(ssid, pass);
LiquidCrystal_PCF8574 lcd_1(0x27);
bool sincronizzato = false;
String stazioni="arpaia";
const int pinRosso = 6;
const int pinVerde = 5;
const int pinRosso1 = 4;
const int pinVerde1 = 2;
Servo servo_1;

WiFiUDP ntpUDP;
//NTPClient timeClient(ntpUDP);
WiFiClient wifi;
HttpClient client = HttpClient(wifi, "10.150.253.251", 80);

void setup() {

```

```

Serial.begin(9600);
servo_1.attach(3);
lcd_1.begin(16, 2);
lcd_1.setBacklight(200);
lcd_1.clear();
lcd_1.print("BENVENUTO ALLA");
lcd_1.setCursor(0, 1);
lcd_1.print("STAZIONE ARPAIA");
delay(1500);
lcd_1.clear();
SPI.begin();
rfid.PCD_Init();

setDebugMessageLevel(2);

WiFi.begin(ssid, pass);

Serial.print("Connessione in corso");
int tentativi = 0;
while (WiFi.status() != WL_CONNECTED && tentativi < 20) {
    delay(1000);
    Serial.print(".");
    tentativi++;
}
delay(1000);
if (WiFi.status() == WL_CONNECTED) {
    Serial.println("\nConnesso!");
    Serial.print("IP assegnato: ");
    Serial.println(WiFi.localIP());
} else {
    Serial.println("\nConnessione fallita. Controlla SSID/Password o Hotspot.");
}
pinMode(pinRosso, OUTPUT);
pinMode(pinVerde, OUTPUT);
pinMode(pinRosso1, OUTPUT);
pinMode(pinVerde1, OUTPUT);
Serial.println("Sincronizzazione stato stallo...");
digitalWrite(pinVerde1, HIGH);
}

void loop() {
    if (!sincronizzato && WiFi.status() == WL_CONNECTED) {
        Serial.println("Sincronizzazione stato stallo...");
        client.get("/stato_stalli.php");
        String risp = client.responseBody();
        int stato = risp.toInt();
    }
}

```

```

digitalWrite(pinVerde, (stato == 1 ? HIGH : LOW));
digitalWrite(pinRosso, (stato == 1 ? LOW : HIGH));
servo_1.write((stato == 1 ? 0 : 90));

lcd_1.setCursor(0, 0);
lcd_1.print("Avvicina una");
lcd_1.setCursor(0, 1);
lcd_1.print("tessera RFID...");

sincronizzato = true;
Serial.println("Sincronizzazione completata.");
}
if (!rfid.PICC_IsNewCardPresent()) {
    return;
}

// Legge la tessera
if (!rfid.PICC_ReadCardSerial()) {
    return;
}
Serial.print("UID della tessera: ");

String uidString = "";
for (byte i = 0; i < rfid.uid.size; i++) {
    uidString += String(rfid.uid.uidByte[i] < 0x10 ? "0" : "");
    uidString += String(rfid.uid.uidByte[i], HEX);
}
uidString.toUpperCase();
uidString.replace(" ", "");
Serial.print("Invio UID: ");
Serial.println(uidString);
Serial.println(stazioni);
client.get("/lettore_bici.php?uid=" + uidString + "&stazioni=" + stazioni);
int statusCode = client.responseStatusCode();
String response = client.responseBody();
if (response.indexOf("Prelevata") > 0) {
    digitalWrite(pinVerde, LOW);
    digitalWrite(pinRosso, HIGH);
    servo_1.write(90);
    lcd_1.clear();
    lcd_1.setCursor(0, 0);
    lcd_1.print("Bici prelevata ");
    Serial.println("AZIONE: Bici prelevata. Stallo ora LIBERO.");
}
else if (response.indexOf("posata") > 0) {
    digitalWrite(pinRosso, LOW);
    digitalWrite(pinVerde, HIGH);
    servo_1.write(0);
}

```

```

        lcd_1.clear();
        lcd_1.setCursor(0, 0);
        lcd_1.print("Bici posata ");
        Serial.println("AZIONE: Bici posata. Stallo ora OCCUPATO.");
    }
    else if (response.indexOf("ERRORE") > 0) {
        for(int i=0; i<3; i++){
            digitalWrite(pinRosso, HIGH); delay(200); digitalWrite(pinRosso, LOW);
delay(200);
        }
    }

    Serial.print("Risposta Server: ");
    Serial.println(response);
    if (response.indexOf("stallo") > 0) {
        int pos = response.indexOf("stallo") + 7;
        String numStallo = response.substring(pos, pos+1);
        delay(1000);
        lcd_1.clear();
        lcd_1.setCursor(0, 0);
        lcd_1.print("Stallo ");
        lcd_1.setCursor(0, 1);
        lcd_1.print("numero: " + numStallo);
        Serial.println("Parcheggia al numero: " + numStallo);
        delay(1000);
        lcd_1.clear();
        lcd_1.setCursor(0, 0);
        lcd_1.print("Avvicina una");
        lcd_1.setCursor(0, 1);
        lcd_1.print("tessera RFID...");
    }

    Serial.print("Status code: ");
    Serial.println(statusCode);
    Serial.print("Risposta: ");
    Serial.println(response);
    rfid.PICC_HaltA();
}

```