

UniBookMi

A.A 2023/2024

Corso di Programmazione Web e Mobile

By

Paolo Faltaous ~ 01828A

Giulia Floris ~ 984706

UniBookMi

Autore: Paolo Faltaous & Giulia Floris

Versione: 1.0.0

Prima modifica: 18/02/2024

Ultima modifica: 19/02/2024



Indice:

[1.0 - Introduzione:](#)

- 1.1 Descrizione
- 1.2 Destinatari
- 1.3 Modello di Lavoro
- 1.4 Flussi di Dati

[2.0 - Interfacce:](#)

[3.0 - Architettura:](#)

- 3.1 Diagramma UniBookMi
- 3.2 Descrizione delle risorse

[4.0 - Codice:](#)

- 4.1 - .HTML
- 4.2 - .CSS
- 4.3 - . JAVASCRIPT
- 4.4 - Database

[5.0 - Conclusioni:](#)

- 5.1 Work in Progress

[6.0 – Note Bibliografiche:](#)

UniBookMi

1. INTRODUZIONE

1.1 Descrizione

UniBookMi è una piattaforma web progettata per semplificare il processo di prenotazione di aule o stanze presso l'Università degli Studi di Milano (Unimi). Gli utenti possono accedere utilizzando il loro account Unimi e prenotare le strutture desiderate tramite un comodo modulo online.

Le principali sezioni dell'applicazione includono:

- Il login
- La homepage
- La pagina Dashboard
- La pagina delle notizie.
- La pagina delle Prenotazioni Aule.
- La pagina dei contatti
- Logout

Con un'interfaccia intuitiva e accessibile, l'applicazione offre a studenti e docenti un modo semplice per navigare e utilizzare le sue funzionalità.

1.2 Destinatari

UniBookMi è stata progettata specificamente per soddisfare le esigenze degli studenti dell'Università Statale di Milano che desiderano prenotare aule per le attività didattiche.

Con un'interfaccia chiara e intuitiva, l'applicazione è accessibile anche a coloro che non hanno conoscenze tecniche approfondite. Ottimizzata per l'uso su computer, sia fissi che portatili, UniBookMi semplifica il processo di prenotazione delle strutture universitarie.

Rivolta principalmente a un pubblico che cerca postazioni di studio o aule. L'applicazione è uno strumento organizzativo essenziale per facilitare la ricerca e la prenotazione degli spazi necessari. La sua motivazione è attiva e diretta, mirando a semplificare e ottimizzare il processo di prenotazione per gli utenti universitari.

1.3 Modello di lavoro

Il fulcro del nostro progetto, UniBookMi, si concentra esclusivamente sulle prenotazioni delle aule per gli studenti dell'Università Statale di Milano. Ci differenziamo da qualsiasi altra piattaforma di prenotazione perché offriamo un ambiente unico e dedicato, senza la necessità di dover cercare servizi simili altrove. Tutto ciò di cui hai bisogno è a portata di clic sul nostro sito.

Prima di iscriverti a una facoltà, sai quanto sia importante avere informazioni dettagliate e precise sulle aule disponibili. Con UniBookMi, ti offriamo la possibilità di prenotare aule in modo efficiente e diretto, risparmiando tempo e fatica. Non c'è bisogno di consultare altre fonti online. Qui troverai tutto ciò di cui hai bisogno per gestire le tue prenotazioni e organizzare al meglio il tuo tempo all'università.

UniBookMi è progettato per semplificarti la vita e per farti sentire parte di una comunità solida e accogliente all'interno dell'Università Statale di Milano.

1.4 Flusso di dati

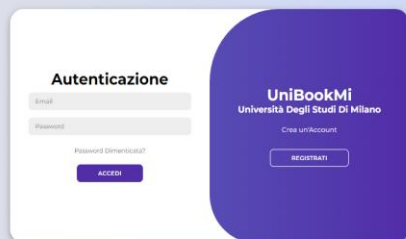
I contenuti sono standardizzati seguendo uno schema uniforme per ciascun dipartimento al quale l'utente è affiliato. Tutti i contenuti visualizzati sono generati utilizzando le informazioni fornite dall'Università Statale di Milano, eliminando così i costi di produzione. I dati vengono archiviati tramite l'utilizzo di un database MySQL, garantendo un'efficace gestione e accessibilità alle informazioni necessarie.

UniBookMi

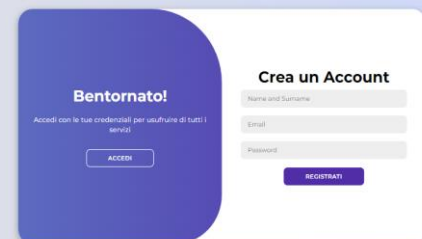
2. INTERFACCE

Al momento dell'accesso al sito, gli utenti vengono indirizzati alla pagina iniziale, chiamata Login.html. Questa pagina offre la possibilità di accedere con le proprie credenziali o di registrarsi per ottenere un account. Dopo aver effettuato l'accesso o la registrazione, gli utenti vengono reindirizzati alla pagina successiva, index.html, che è la homepage di UniBookMi.

La pagina di login presenta un'interfaccia accattivante e user-friendly, con un comodo switch che consente agli utenti di passare facilmente tra la sezione di login e quella di registrazione, e viceversa. Questo permette agli utenti di gestire il proprio accesso al sito in modo intuitivo e semplice.



Attiva Windows
Per la configurazione per attivare Windows.

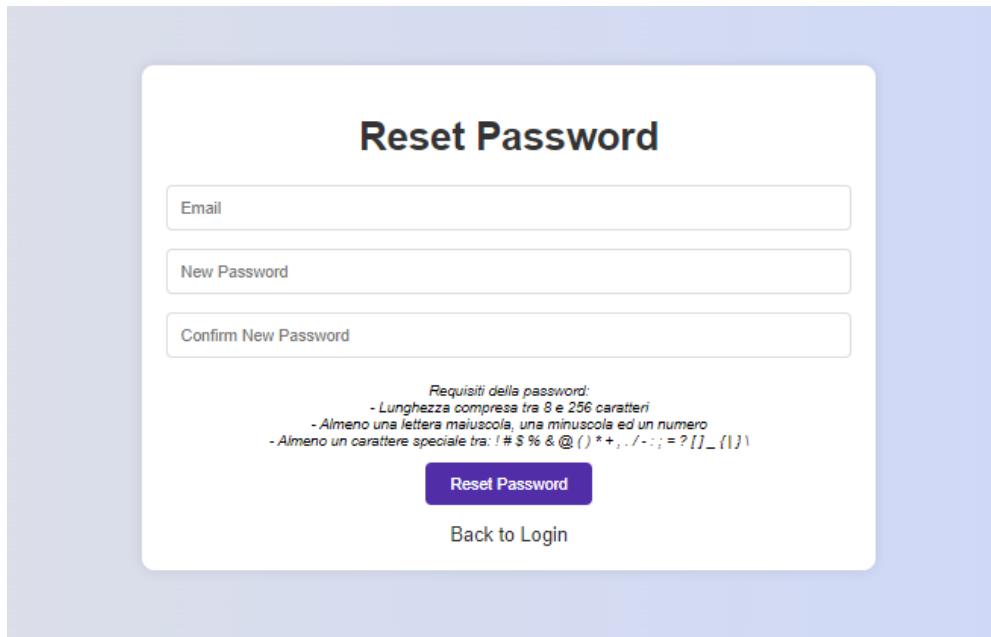


Attiva Windows
Per la configurazione per attivare Windows.

Dopo aver effettuato l'accesso e raggiunto la pagina Index.html, si può facilmente notare la coerenza nel design, mantenendo gli stessi colori e sfondi utilizzati nella pagina di login. Il focus principale della homepage è chiaramente la possibilità di effettuare una prenotazione dell'aula "Prenota Qui!", che rappresenta il motivo principale per cui è stata sviluppata questa applicazione web.

Inoltre, si può apprezzare uno sfondo dinamico in movimento che mira a evidenziare in modo dettagliato il servizio offerto dal sito. Questo elemento visivo aggiunge dinamicità e attrattiva alla pagina principale, catturando l'attenzione degli utenti e facilitando la comprensione immediata delle funzionalità offerte da UniBookMi.

Nella pagina di login, qualora l'utente si fosse dimenticato la password, ha la possibilità di resettarla cliccando "Password Dimenticata"



Reset Password

Requisiti della password:
 - Lunghezza compresa tra 8 e 256 caratteri
 - Almeno una lettera maiuscola, una minuscola ed un numero
 - Almeno un carattere speciale tra: ! # \$ % & @ () * + , . / - : ; = ? [] _ { | } \

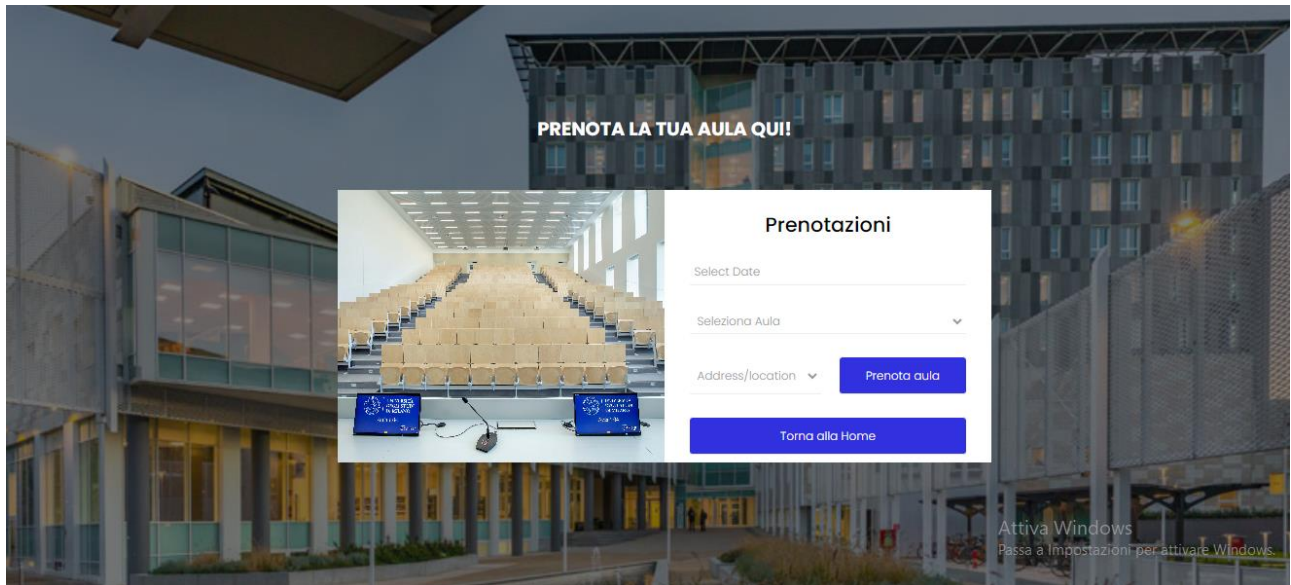
[Back to Login](#)

Esaminando la Homepage, è evidente la presenza delle sezioni menzionate nella [1.1 Descrizione](#) precedente, collocate in alto a destra. Queste sezioni permettono agli utenti di utilizzare il sito web in modo completo e approfondito, fornendo accesso rapido alle varie funzionalità offerte da UniBookMi.

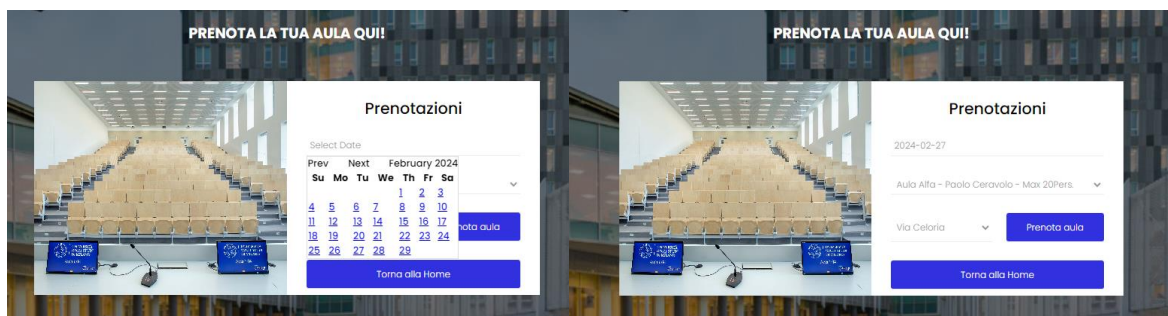


Dopo aver cliccato sul pulsante "Prenota Qui!" o "Prenotazione Aule" situato in alto a destra, saremo immediatamente reindirizzati alla pagina dedicata che ci consente di effettuare la prenotazione dell'aula desiderata.

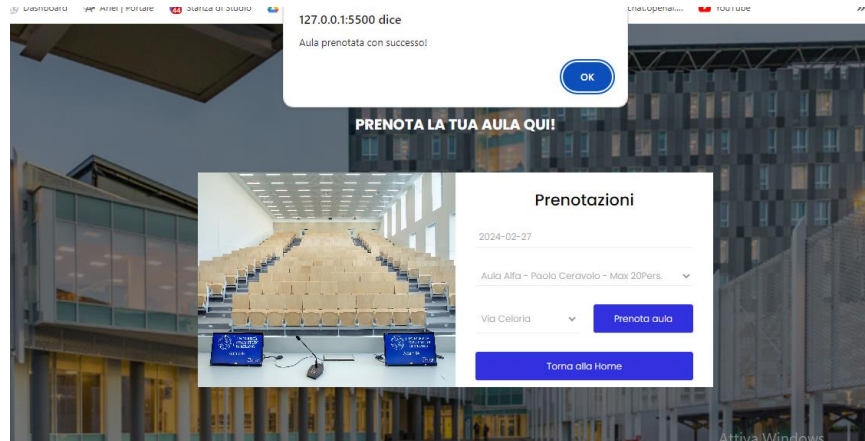
Qui potremo specificare la data, l'aula e l'edificio preferiti, fornendo agli utenti un accesso diretto e intuitivo alla funzione di prenotazione delle aule.



La coerenza nel design è evidente, con l'utilizzo dei medesimi colori e sfondi sia nella pagina di login che nella homepage. Inoltre, lo sfondo raffigura l'edificio universitario situato in Via Celoria 18, aggiungendo un elemento distintivo e riconoscibile alla pagina di prenotazione delle aule.



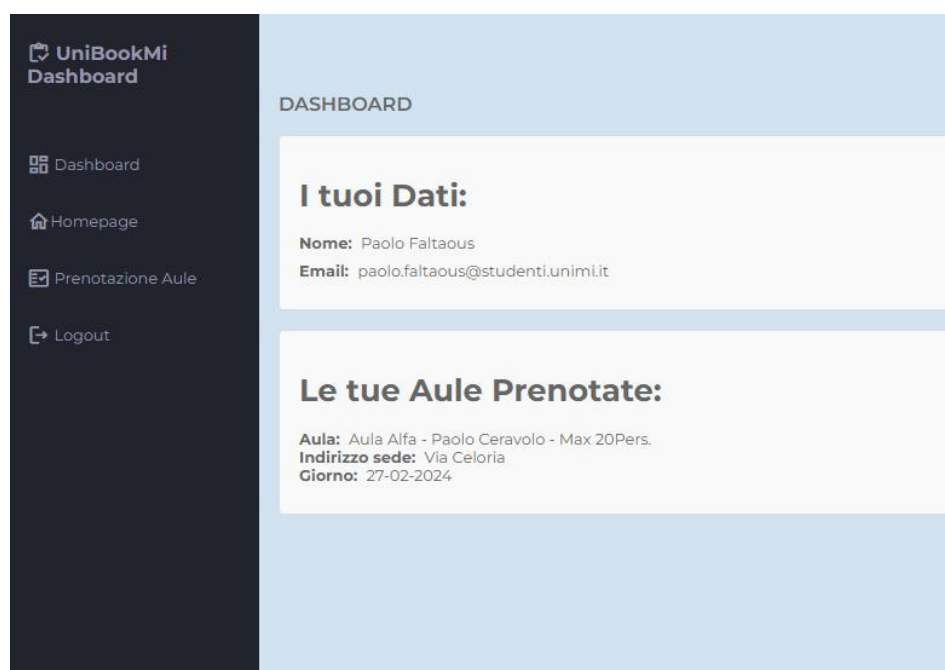
Dopo aver premuto il pulsante "Prenota Aula", comparirà un messaggio che conferma il successo della prenotazione con il testo "Aula prenotata con successo".



Tutti i dettagli della prenotazione verranno salvati nella tabella "Reservation" del database dedicato di UniBookMi. Tuttavia, i dettagli su come questo processo sia implementato sono trattati nel capitolo [4.0 Codice](#)

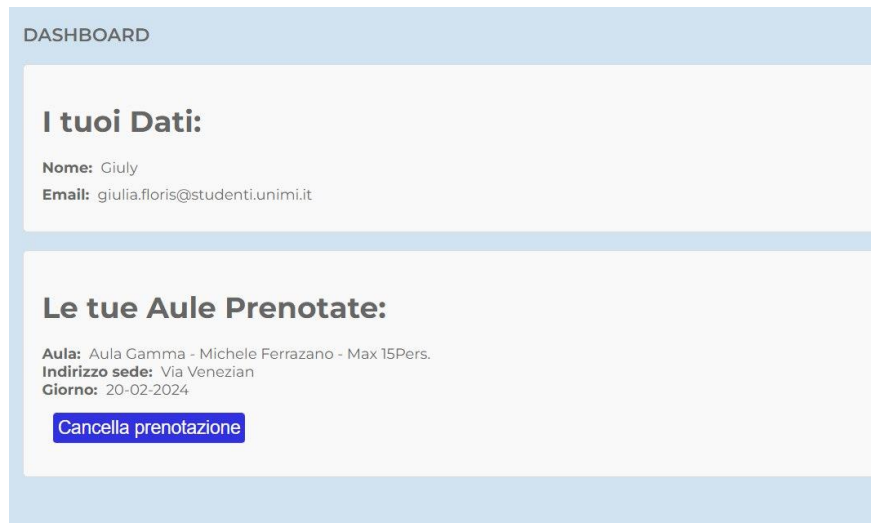
Dopo aver confermato la prenotazione cliccando su "ok" al messaggio di conferma, saremo automaticamente reindirizzati alla Homepage.

Successivamente, accedendo alla sezione "Dashboard", gli utenti possono trovare un riassunto dei propri dati e delle prenotazioni effettuate.

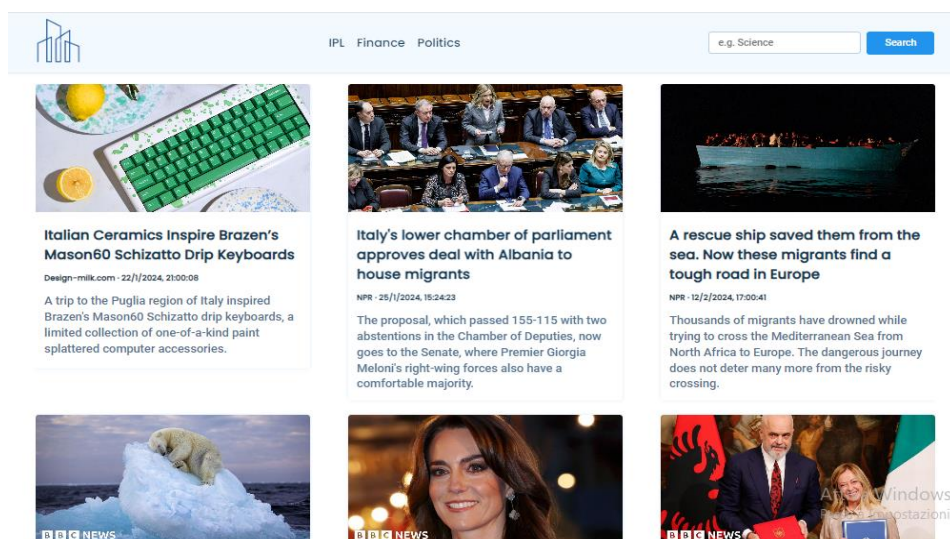


Il sito web mantiene una coerenza visiva attraverso tutte le pagine, utilizzando le stesse colorazioni ad eccezione della barra laterale. Questa barra laterale raggruppa le diverse sezioni presenti nella Homepage, facilitando la navigazione e rendendo l'applicativo web più efficiente negli spostamenti tra le varie pagine.

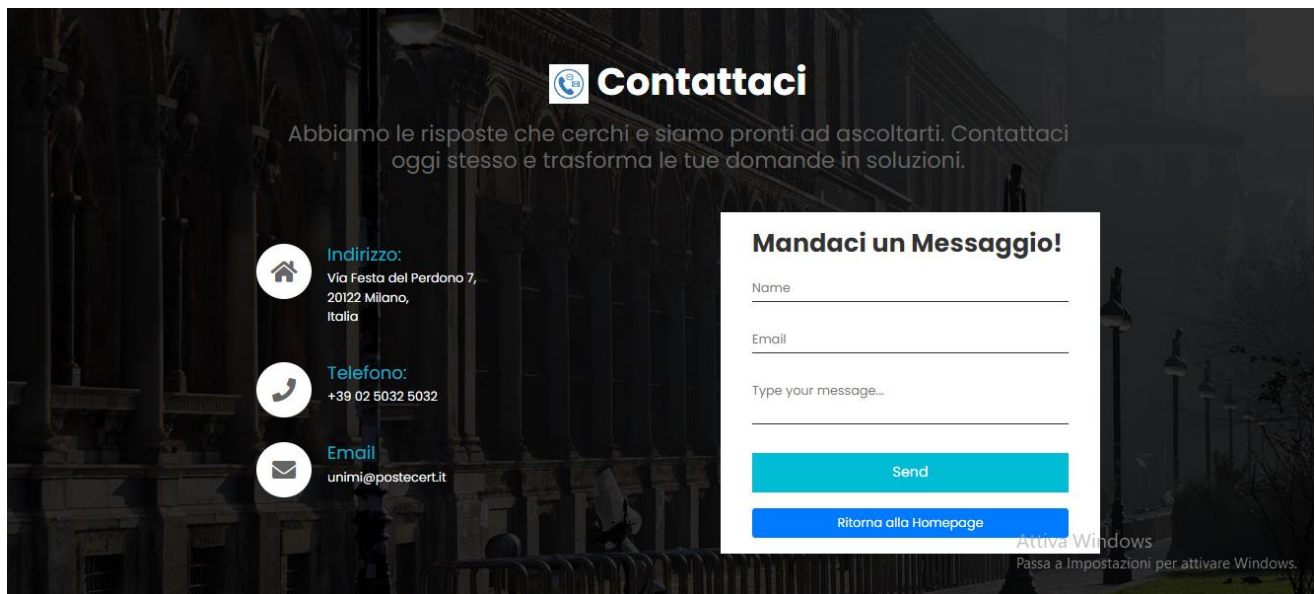
È inoltre possibile effettuare l'eliminazione della prenotazione tramite apposito tasto presente nella sezione "Le tue aule prenotate"



Una volta tornati alla Homepage, gli utenti possono visitare la pagina delle News per rimanere aggiornati sulle ultime notizie riguardanti generiche e di COVID-19 in tutto il mondo. Questo è uno dei motivi principali per cui è stato sviluppato l'applicativo web.



Inoltre, è possibile accedere alla pagina dei Contatti, dove gli utenti possono trovare i dettagli dell'Università Statale di Milano e inviare un messaggio compilando i campi relativi.



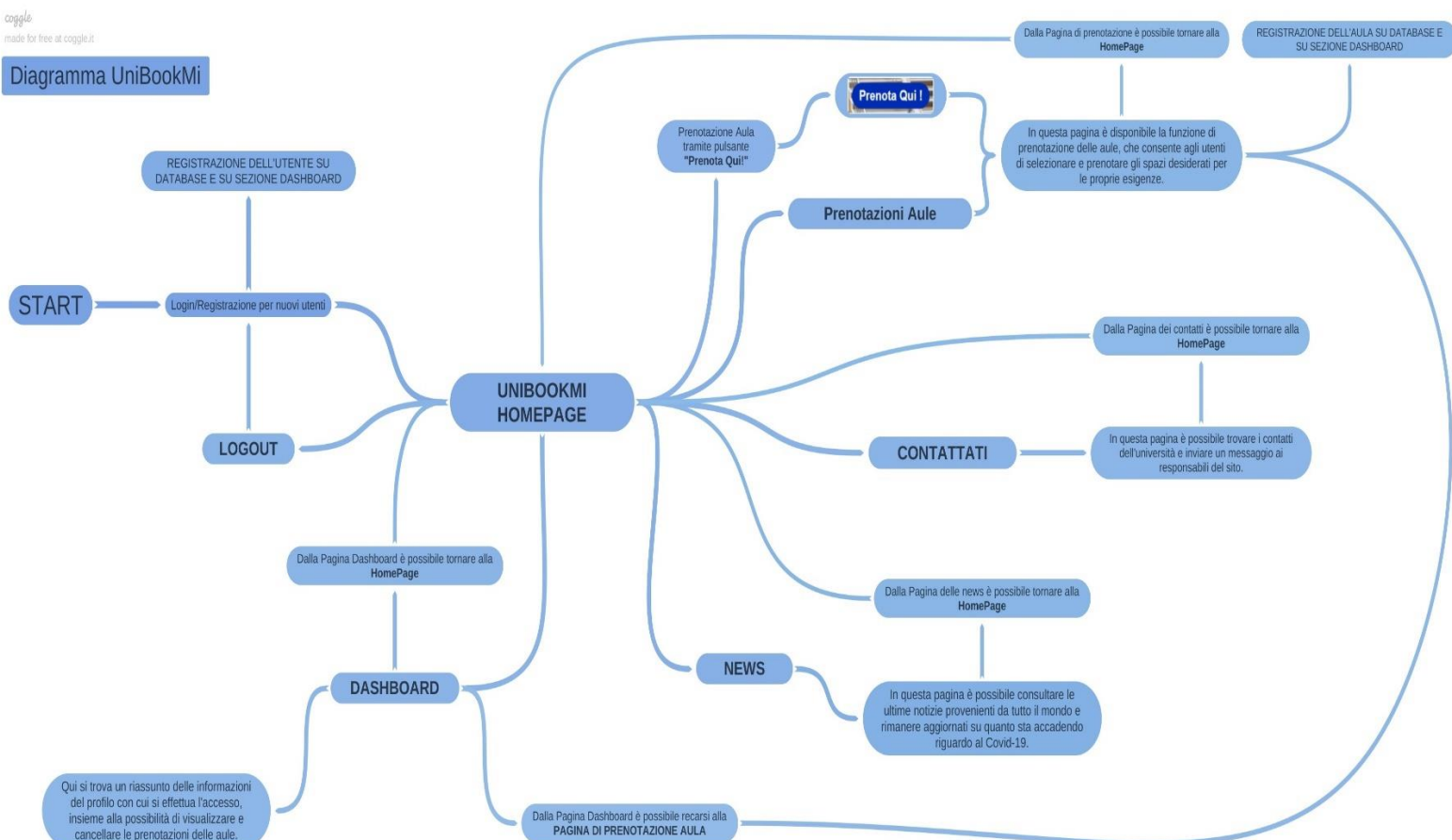
Come si può notare anche qui è presente uno sfondo che rappresenta la sede principale dell'Università Statale di Milano.

UniBookMi


















3. ARCHITETTURA







3.1 Diagramma UnibookMi

Di seguito è riportato lo schema gerarchico dell'applicativo Web UniBookMi



3.2 Descrizione delle risorse

-  [Server.js](#) – File configurazione server
-  [user.js](#) – File contenete logica del server e gestione delle chiamate al DB
-  [Login.html](#) – Pagina di Login
-  [login_script.js](#) – Script della pagina login
-  [login.css](#) – Foglio di stile per pagina di login
-  [Resetpassword.html](#) – Pagina di ResetPassword
-  [Reset_password.js](#) – Script della pagina ResetPassword
-  [Resetpassword.css](#) – Foglio di stile per pagina di login
-  [Index.html](#) – Homepage principale
-  [main.css](#) – Foglio di stile della pagina principale
-  [mainScript.js](#) – Script della pagina HomePage
-  [Reservation.html](#) – Pagina di Prenotazioni Aule
-  [Reservation.css](#) – Foglio di stile della Prenotazioni Aule
-  [Reservation.js](#) – Script della pagina Prenotazioni Aule
-  [Dashboard.html](#) – Pagina di Dashboard riepilogativa
-  [userinfo.css](#) – Foglio di stile della pagina Dashboard
-  [Dashboard.js](#) – Script della pagina Dashboard

-  [NewsView.html](#) – Pagina di News
-  [News.css](#) – Foglio di stile della pagina News
-  [NewsScript.js](#) – Script della pagina News
-  [Contacts.html](#) – Pagina con i Contatti
-  [contacts.css](#) – Foglio di stile della pagina Contatti
-  [colors.css](#) – Foglio stile per scelta dei colori del sito

UniBookMi

4. CODICE

4.1 HTML

```
<body>
  <div class="container" id="container">
    <div class="form-container sign-up">
      <form id="formRegister" >
        <h1>Crea un Account</h1>
        <input type="text" placeholder="Name and Surname" name="nome", id="nomeRegistration">
        <input type="email" placeholder="Email" name="email", id="emailRegistration">
        <input type="password" placeholder="Password" name="password", id="passwordRegistration">
        <button type="submit" id="signUp">Registrati</button>
      </form>
    </div>
    <div class="form-container sign-in">
      <form>
        <h1>Autenticazione</h1>
        <input type="email" placeholder="Email", id="emailLogin">
        <input type="password" placeholder="Password" id="passwordLogin">
        <a href="http://127.0.0.1:5500/features/login/resetpassword.html">Password Dimenticata?</a>
        <button id="signIn">Accedi</button>
      </form>
    </div>
    <div class="toggle-container">
      <div class="toggle">
        <div class="toggle-panel toggle-left">
          <h1>Bentornato!</h1>
          <p>Accedi con le tue credenziali per usufruire di tutti i servizi</p>
          <button class="hidden" id="login">Accedi</button>
        </div>
        <div class="toggle-panel toggle-right">
          <h1>UniBookMi</h1>
          <h3>Università Degli Studi Di Milano</h3>
          <p>Crea un'Account</p>
          <button class="hidden" id="register">Registrati</button>
        </div>
      </div>
    </div>
  </div>

  <!-- script always goes under all html code-->
  <div>
    <script src="http://127.0.0.1:5500/features/login/login_script.js"></script>
  </div>
</body>
```

Uno tra i frammenti più importanti troviamo l'area contenente il form per il login/registrazione.

La prima sezione contiene un modulo per la registrazione con campi per il nome, l'e-mail e la password, insieme a un pulsante "Registrati".

La seconda sezione contiene un modulo per l'autenticazione (accesso) con campi per l'e-mail e la password, insieme a un pulsante "Accedi" e un link per ripristinare la password.

La terza sezione è un interruttore che consente di passare tra le modalità "Accedi" e "Registrati", mostrando i relativi messaggi e pulsanti.

Sotto queste sezioni, è incluso un riferimento a uno script JavaScript esterno login_script.js per gestire le funzionalità dinamiche della pagina, come la validazione dei dati e la gestione degli eventi.

Nella pagina Login_script.js è contenuto un piccolo script che permette di gestire un modulo di registrazione e accesso su una pagina web.

Quando un utente clicca sul pulsante "register", viene mostrato il modulo di registrazione. Se l'utente inserisce i dati correttamente e fa clic su "signUp", i dati vengono inviati al server tramite una richiesta POST e l'utente viene reindirizzato a una pagina di home.

Similmente, per il login, i dati vengono inviati al server tramite una richiesta GET, e se l'autenticazione ha successo, l'utente viene reindirizzato alla pagina di home.

4.2 CSS

```
body {
  background-color: #c9d6ff;
  background: linear-gradient(to right, #e2e2e2, #c9d6ff);
  display: flex;
  align-items: center;
  justify-content: center;
  flex-direction: column;
  height: 100vh;
}
```

```
.container {
  background-color: #fff;
  border-radius: 30px;
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.35);
  position: relative;
  overflow: hidden;
  width: 868px;
  max-width: 100%;
  min-height: 480px;
}
```

```
.form-container {
  position: absolute;
  top: 0;
  height: 100%;
  transition: all 0.6s ease-in-out;
}
```

```
.toggle-container {
  position: absolute;
  top: 0;
  left: 50%;
  width: 50%;
  height: 100%;
  overflow: hidden;
  transition: all 0.6s ease-in-out;
  border-radius: 150px 0 0 100px;
  z-index: 1000;
}
```

```
.toggle-left {
  transform: translateX(-200%);
}

.container.active .toggle-left {
  transform: translateX(0);
}

.toggle-right {
  right: 0;
  transform: translateX(0);
}

.container.active .toggle-right {
  transform: translateX(200%);
}
```

Il codice CSS definisce una pagina web con un layout flessibile e responsivo per un form di accesso (sign-in) e registrazione (sign-up). Ecco una breve panoramica delle parti più importanti:

- **Import dei font:** Importa il font "Montserrat" da Google Fonts con vari pesi (300, 400, 500, 600, 700) per essere utilizzato nel documento.
- **Reset del CSS e definizione del font predefinito:** Imposta lo stile di base del documento, rimuovendo margini, padding e impostando il font di default a "Montserrat".
- **Stile del body:** Imposta uno sfondo con un gradiente lineare e centra il contenuto verticalmente e orizzontalmente nella pagina.
- **.container:** Questa classe definisce lo stile del contenitore principale che tiene il form di accesso e registrazione. Ha un background bianco, un bordo arrotondato e una leggera ombra. Inoltre, ha una larghezza massima del 100% della finestra del browser e una altezza minima di 480px.
- **Transizioni e animazioni:** Utilizza le transizioni e le animazioni CSS per creare effetti visivi durante il passaggio tra le modalità di accesso e registrazione.
- **.form-container:** Questa classe gestisce l'animazione di transizione tra la modalità di accesso e registrazione.
- **Toggle tra modalità di accesso e registrazione:** Utilizza un toggle per passare tra il form di accesso e il form di registrazione, con un effetto di dissolvenza e scorrimento.
- **Stili dei componenti del form:** Definisce lo stile degli input, dei pulsanti e dei link all'interno dei form di accesso e registrazione.

4.3 JAVASCRIPT

Nel file User.js sono contenute alcune delle stringhe più importanti del progetto che permettono l'utilizzo dei vari servizi. Inizialmente, vengono importati i moduli necessari, tra cui mysql2 per la connessione al database MySQL, bcrypt per la crittografia delle password, express per la gestione del server HTTP, cors per abilitare la condivisione delle risorse tra origini diverse e body-parser per l'analisi dei corpi delle richieste in ingresso.

```
const mysql = require('mysql2');
const bcrypt = require('bcrypt');
const express = require('express');
const cors = require('cors');
const bodyParser = require('body-parser');

var userEmail = "";
var userName = "";

const app = express();
// Enable CORS for all routes
app.use(cors());
const port = process.env.PORT || 3000;

// Middleware to parse incoming request bodies
app.use(bodyParser.urlencoded({ extended: true }));

app.use(express.json());

var con = mysql.createConnection({
  host: "127.0.0.1",
  user: "root",
  password: "Paolo2002-",
  database: 'unibookmi'
});

con.connect(err => {
  if (err) {
    console.error('Error connecting to MySQL database:', err);
    return;
  }
  console.log('🔥🔥 Connected to MySQL database');
});

app.listen(port, () => {
  console.log('🔥🔥 Server is running on http://localhost:${port}');
});

app.get('/getInfo', (req, res) => {
  res.json({ name: userName, email: userEmail });
});
```

Successivamente, viene creato un'app Express e configurato il middleware. Il *middleware cors()* abilita CORS per tutte le rotte, mentre *bodyParser.urlencoded({ extended: true })* e *express.json()* sono utilizzati per analizzare i corpi delle richieste con dati codificati URL e JSON, rispettivamente.

Viene quindi stabilita la connessione al database MySQL utilizzando le credenziali specificate. In caso di successo, viene stampato un messaggio di connessione riuscita; in caso di errore, viene visualizzato un messaggio di errore.

Il server Express viene quindi avviato sulla porta specificata, e viene definito un endpoint HTTP GET /getInfo. Quando viene effettuata una richiesta a questo endpoint, il server risponde restituendo le informazioni dell'utente memorizzate nelle variabili userName e userEmail in formato JSON.

Questo codice è un endpoint POST per un servizio che consente di aggiungere un utente al database.

```
// POST add user service
app.post('/addUser', (req, res) => {

  console.log("⚡⚡ POST BODY -> ", req.body)
  const { email, nome, password } = req.body;
  var sql = "INSERT INTO utenti (email, nome, password) VALUES (?, ?, ?)";
  bcrypt.hash(password, 10, (err, hash) => {
    if (err) {
      console.error('Error hashing password:', err);
      return;
    }
    console.log('Hashed password:', hash);
    con.query(sql, [email, nome, hash], (err, result) => {
      if (err) {
        console.error('💀💀 Error executing INSERT query:', err);
        res.status(500).send('Error executing INSERT query:');
        return;
      } else {
        userName = nome;
        userEmail = email;
        res.status(200).json({ message: 'Insert successful' });
        console.log('✅✅ INSERT query successful');
      }
    });
  });
});
```

Quando riceve una richiesta POST su /addUser, il server estrae i dettagli dell'utente (e-mail, nome e password) dal corpo della richiesta. Successivamente, la password viene criptata utilizzando l'algoritmo di hash bcrypt con un costo di 10 iterazioni.

Successivamente, viene eseguita una query SQL per inserire i dati dell'utente criptati nel database. Se l'inserimento ha successo, viene restituita una risposta con stato 200 e un messaggio JSON che indica il successo dell'operazione. In caso di errore durante l'esecuzione della query, viene restituito un messaggio di errore con stato 500.

Un'altra sezione importante è la seguente:

```
// GET all reservations
app.get('/getReservations', (req, res) => {

  const sql = `SELECT * FROM reservations WHERE email = ?`;
  con.query(sql, [userEmail], (err, result) => {
    if (err) {
      console.error('💀💀Error executing SELECT query:', err);
      res.status(500).json({ message: `Nessuna prenotazione con email ${email}` });
      throw err;
    } else {
      console.log("RESULT -> ", result[0]);
      res.json(result);
      console.log('✅✅ SELECT query successful');
    }
  });
});
```

Definisce un endpoint GET per un servizio che recupera le prenotazioni di un utente dal database.

Quando viene inviata una richiesta GET a /getReservations, il server esegue una query SQL per selezionare tutte le prenotazioni corrispondenti all'email specificata.

```
con.query(sql, [userEmail], (err, result) => {
  if (err) {
    console.error('💀💀Error executing SELECT
query:', err);
    res.status(500).json({ message: `Nessuna
prenotazione con email ${email}` });
    throw err;
  } else {
    console.log("RESULT -> ", result[0]);
    res.json(result);
    console.log('✅✅ SELECT query successful');
  }
});
```

In questa parte, viene eseguita la query SQL utilizzando l'e-mail dell'utente per filtrare le prenotazioni corrispondenti nel database. Se si verifica un errore durante l'esecuzione della query, viene restituito un messaggio di errore con stato 500 e il flusso del programma viene interrotto con throw err. Se non ci sono errori, le prenotazioni trovate vengono inviate come risposta JSON. Infine, viene stampato un messaggio di log per indicare che la query SELECT è stata eseguita con successo.

4.4 Database

Il database è stato sviluppato tramite MySQL e comprende due tabelle principali: "Utentii" e "Reservation". Queste tabelle sono fondamentali per le operazioni di recupero e inserimento dei dati.

La tabella "Reservation" è dedicata alla registrazione delle prenotazioni delle aule da parte degli studenti. Essa fornisce anche un meccanismo per verificare lo stato delle prenotazioni effettuate.

La tabella "Utentii", invece, gestisce le informazioni sugli utenti registrati nel sistema. Viene utilizzata per verificare le credenziali durante il processo di login, garantendo che corrispondano alle informazioni dell'utente registrato.

L'obiettivo principale è stato rendere la gestione del database il più intuitiva possibile, agevolando sia la creazione che la consultazione dei dati relativi alle prenotazioni delle aule e agli utenti del sito.

Tabelle Utentii e Reservation:

The screenshot shows a database management interface with the 'Utentii' table selected. The table structure is as follows:

email	Nome	Password
Giuliaforis@gmail.com	Giulia	\$2b\$10\$Xa0Z.SC5H9FuLP7NGTP.M5VgasD6kCRmwO7POd...
irini@gmail.com	Irini Faltaous	\$2b\$10\$YwM.KxqBNyNwdnrWVg8f.QCPmxID.9Cxt6sCpNC...
irini2@gmail.com	irini	\$2b\$10\$Yab0t7EIPQjx4FcgYNOne09hayAud3xQ5Kv6NSC...
KirolesFaltaous@gmail.com	Kiro Faltaous	\$2b\$10\$HsjeGhGxwJ/2JN92qn1zw.Ep380CawFEu9DNr9y1...
paolo.faltaous@studenti.unimi.it	Paolo Faltaous	\$2b\$10\$Cn8F1cTdd.XsWDPq2A.6.bum4I/cuHfy49.bOon2...
paolo2@gmail.com	paolo2	\$2b\$10\$CvjuYGV0xYv3npUHzA.epXZ70FBVHVHWowbEFv8...
paolofaltaous@gmail.com	paolo	\$2b\$10\$M0Hc7dirO.dqh78CQfju3+HAls9vLZOyoW01/N3Mr...
paolofaltaous29@gmail.com	Paolo Faltaous	\$2b\$10\$Hk4Nxm1cpvFsl.qxZ1.8.mltafyD8sE.o/2x8SbjO4m...
paolofaltaousss@gmail.com	Paolo Faltaous	\$2b\$10\$XFLtclUMaFabK1BYnmFhm7eBPrQCXsisF3I2eDffYL...
NULL	NULL	NULL

The screenshot shows a database management interface with the 'Reservation' table selected. The table structure is as follows:

email	date	address	roomType
paolofaltaousss@gmail.com	2024-02-20	Via Venezian	Aula Gamma - Michele Ferrazano - Max 15Pers.
paolo.faltaous@studenti.unimi.it	2024-02-27	Via Celoria	Aula Alfa - Paolo Ceravolo - Max 20Pers.
NULL	NULL	NULL	NULL

UniBookMi

5. CONCLUSIONI

5.1 WORK IN PROGRESS

Il progetto ha dimostrato la fattibilità di sviluppare un'applicazione web per la gestione del registro di accesso a un'Aula utilizzando le tecnologie indicate precedentemente.

Lo studio di fattibilità e la ricerca costante hanno contribuito al successo del progetto.

L'applicazione web sviluppata offre una soluzione pratica e conforme alle normative vigenti per la gestione delle prenotazioni delle aule.

UniBookMi

6. NOTE BIBLIOGRAFICHE

Fonti:

- How to connect Microsoft SQL Server with Node js Application – <https://www.youtube.com/watch?si=bvywsuHyUDAcspEM&v=9cmv9deKWHw&feature=youtu.be>
- MySQL Database connection from node js application - <https://www.youtube.com/watch?v=eljbSH3lmb8>
- REST API validation in nodejs – https://www.youtube.com/watch?v=F-1GD_F8jHg&t=5s
- How To Create Working Contact Form Using HTML & CSS | Receive Contact Form Data on Email – <https://www.youtube.com/watch?v=-HeadgoqJ7A>
- How to create a website login HTML, CSS, JAVASCRIPT– <https://www.youtube.com/@codehal>
- How to create a website reservation with HTML, CSS, JAVASCRIPT – <https://www.youtube.com/@geekprobin1456>
- Come scaricare e installare MySQL su Windows 10 ITA – <https://www.youtube.com/watch?v=UH9BUkewzLO>
- Come installare MySQL su Windows [SQL TUTORIAL] – <https://www.youtube.com/watch?v=lKMm-BQnJVE>
- How To Install MySQL (Server and Workbench) - <https://www.youtube.com/watch?v=u96rVINbAUl>
- How to Create a Database, Add Tables and Import Data in MySQL Workbench - <https://www.youtube.com/watch?v=OnXB3ZRrOW0&t=158s>
- SQL Tutorial - 12: Inserting Data Into Tables - <https://www.youtube.com/watch?v=Tet3Z7Yb2gg>

- Learn How to Create a Database | First Steps in SQL Tutorial -
https://www.youtube.com/watch?v=PPFyoXA_FC0
- Connect to MySQL Database from Visual Studio Code and Run SQL
Queries using SQLTools Extension -
<https://www.youtube.com/watch?v=wzdCpJY6Y4c>

Autori:

- Paolo Faltaous
- Giulia Floris