

# Traccia d'Esame - Progetto di sistema informativo per la gestione tornei di Kombat Inc.

Tafuro Alessandro 033400086

L'associazione sportiva Kombat Inc. necessita di un software per la gestione di tornei.

Il software deve prevedere l'utilizzo di un database relazionale per immagazzinare dati relativi a tornei di: MMA, K1, BJJ.

Ad ogni torneo possono essere associate una o più gare/incontri, ad ogni incontro possono partecipare minimo zero massimo due atleti.

C'è la possibilità per un torneo di rendere le iscrizioni a pagamento, il software dovrà gestire anche questa eventualità tramite ticket che verranno poi pagati in una cassa fisica.

Gli utenti dovranno avere la possibilità di accedere alla loro area personale per visualizzare e modificare i dati relativi ai campi: email, password, nome, cognome, numero tessera, grado/cintura, categoria di peso (ove possibile).

Un Torneo è caratterizzato da:

- Nome
- Grado minimo richiesto
- luogo
- data
- podio
- Disciplina

Un Incontro è caratterizzato da:

- Angolo Blu
- Angolo Rosso
- ID
- Esito (angolo blu o angolo rosso)
- Ora
- Categoria
- Coach all'angolo dell'atleta

Gli utenti dell'applicazione si dividono in: organizzatori, coach, atleti, arbitri.

Gli Organizzatori sono caratterizzati da:

- Nome
- Cognome
- email
- Federazione di appartenenza
- Numero di tessera

I Coach sono caratterizzati da:

- Nome
- Cognome
- Palestra di appartenenza
- email
- Numero di tessera
- grado/cintura

I coach possono avere un grado che va da un minimo di 1 dan, ad un massimo di 5 dan.

Gli Atleti sono caratterizzati da:

- Nome
- Cognome
- Numero di tessera
- Palestra di appartenenza
- grado/cintura
- categoria di peso
- storico incontri

I colori di una cintura dell'atleta possono variare tra: bianca, gialla, blu, marrone, nera.

Gli Arbitri sono caratterizzati da:

- Nome
- Cognome
- Email
- Numero di tessera
- Età

Il software deve permettere la registrazione di nuovi atleti e coach e il login tramite email e password.

Un organizzatore può partecipare all'organizzazione di una o più gare, un coach può iscrivere zero o più atleti ad un torneo e un atleta può partecipare a più gare anche nello stesso giorno.

Un Ticket dovrebbe indicare le seguenti informazioni:

- ID
- Costo
- Data prenotazione
- Scadenza

## **Obiettivi del Progetto**

### **1. Analisi e progettazione concettuale**

A partire dalla descrizione del contesto, analizza il dominio e costruisci un modello entità-relazione completo, includendo eventuali relazioni di generalizzazione/specializzazione tra

entità. Spiega in modo chiaro come hai gestito tali relazioni e motiva le scelte effettuate.

## 2. Progettazione logica

Deriva il modello logico relazionale dal modello E-R, includendo una descrizione scritta dei vincoli rilevanti e giustificando eventuali semplificazioni o adattamenti.

## 3. Implementazione del sistema informativo

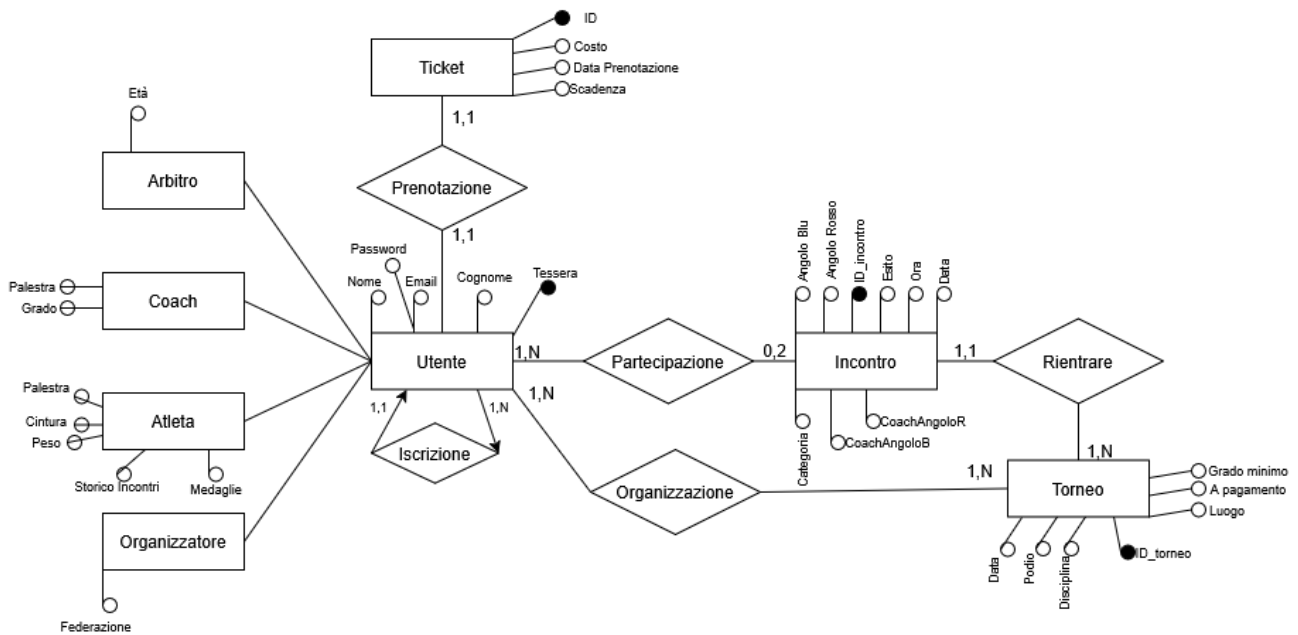
Utilizzando Django, realizza un'applicazione che permetta la gestione dei dati modellati e l'accesso ad almeno quattro funzionalità tra quelle previste nella descrizione iniziale. Tra le funzionalità realizzabili, si possono includere:

- Registrazione e autenticazione degli utenti (differenziati per ruolo).
  - Creazione di nuovi tornei per gli organizzatori
  - Visualizzazione dei dati dei vari incontri
  - Iscrizione degli atleti ai tornei per i coach
  - Visualizzazione e modifica dei dati personali per ogni tipologia di utente
  - Visualizzazione dei dati di un torneo singolo
  - Visualizzazione e download dei ticket da parte dell'atleta
  - Homepage personalizzata per ogni tipologia di utente
  - Creazione dei brackets (degli incontri del torneo) automatizzata
- L'applicazione deve essere realizzata utilizzando esclusivamente:
- Backend Django,
  - Sistema di template di Django per la generazione delle pagine,
  - Bootstrap CSS per la parte grafica.

L'uso di JavaScript è da limitare ai soli casi in cui sia indispensabile.

## Progettazione Concettuale

Un primo schema Entità-Relazione può essere disegnato in questo modo:



Si noti la gerarchia di tipo totale e disgiunta: un utente può rientrare solo in una di queste categorie, e tutte le categorie messe insieme formano tutti i tipi di utente possibili sulla piattaforma.

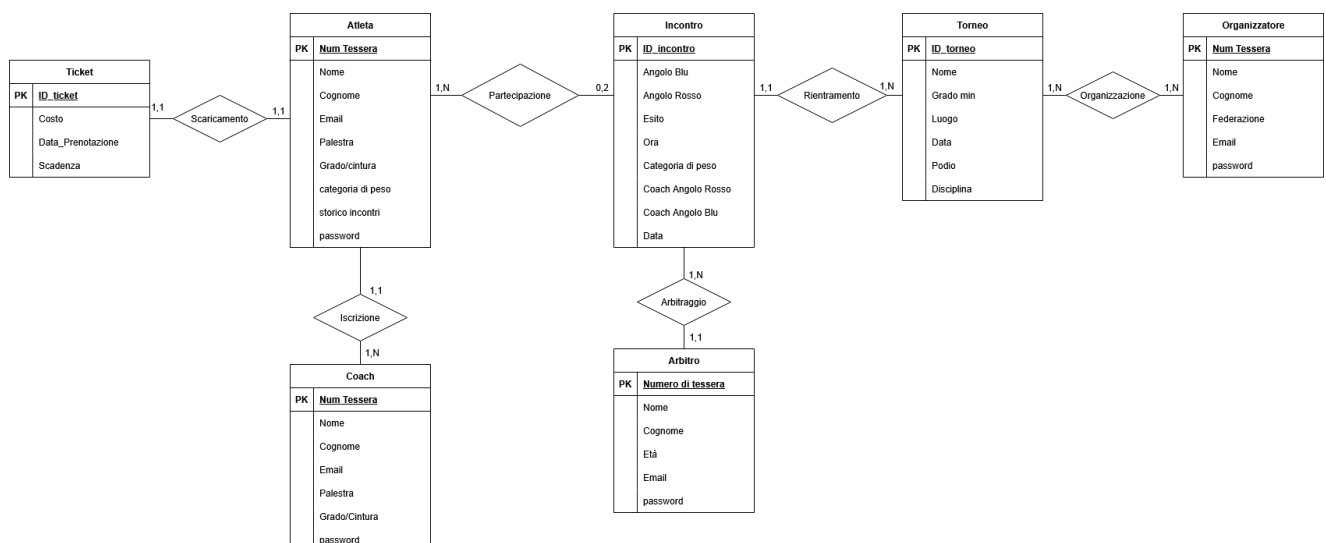
Si è scelto di risolvere la gerarchia creando un'entità separata per ogni entità figlia, facendole ereditare tutti gli attributi dell'entità padre.

Si noti, inoltre, la relazione ricorsiva derivante dal fatto che solo un coach può iscrivere un atleta ad un torneo. Questa relazione particolare viene risolta in automatico andando a separare tutte le entità figlie come descritto in precedenza, così da poter mettere l'entità Coach in diretta relazione con l'entità Atleta.

Un'altra conseguenza della risoluzione della gerarchia è che la relazione tra le entità Torneo e Utente diventa una relazione tra la nuova entità Organizzatore e Torneo.

Le entità Atleta ed Arbitro ereditano la relazione con l'entità Incontro, in due relazioni tra di loro indipendenti.

Lo schema appena descritto appare in questo modo:



# Modello Logico

**Atleta** ( NumTessera, nome, password, cognome, email, palestra, cintura, categoria, storico:Incontro )

**Incontro** ( ID\_incontro, angoloBlu, angoloRosso, esito, ora, categoria, coachAngoloR, coachAngoloB )

**Torneo** ( ID\_torneo, nome, gradoMin, luogo, data, podio, disciplina, a\_pagamento, id\_incontro:Incontro )

**Organizzatore** ( numTessera, nome, password, cognome, federazione, email )

**Ticket** ( ID\_ticket, costo, data\_prenotazione, scadenza, numTessera\_atleta:Atleta )

**Arbitro** ( numTessera, nome, password, cognome, email, età, email, id\_incontro:Incontro )

**Coach** ( numTessera, nome, password, cognome, email, palestra, grado, numTessera\_atleta:Atleta )

**Organizzazione** ( numTessera, ID\_torneo )

## Incontro

Nome Campo	Descrizione	Tipo	Lunghezza	Vincoli
ID_incontro	identificativo univoco dell'incontro	Intero, Autoincrement		Primary Key, unique, not null
AngoloB	Nome atleta all'angolo blu	Varchar	150	Null
AngoloR	Nome atleta all'angolo rosso	Varchar	150	Null
Esito	Indica il vincitore tra i due angoli	Enum	1	Not Null, valori ammessi: ['B' o 'R']
Ora	Ora dello svolgimento dell'incontro	Time		not null
Data	Data dello svolgimento dell'incontro	Date		not null
Categoria	Categoria di peso dell'incontro	Varchar	150	not null
CoachAngoloB	Coach dell'atleta all'angolo Blu	Varchar	150	null
CoachAngoloR	Coach dell'atleta all'angolo Rosso	Varchar	150	null

Si è scelto di rendere "Esito" un attributo Enum per obbligarlo a prendere solo i valori R o B per indicare rispettivamente la vittoria dell'angolo rosso o blu.

Il campo "Categoria" non è numerico per semplificare la stampa del valore dello stesso.

Si noti che non ci sono campi unique se non la chiave primaria, poiché si possono verificare anche più incontri nello stesso momento e più atleti e coach possono gareggiare in più incontri.

I due valori "angoloB" e "angoloR" possono essere null poiché si può verificare la situazione per cui un incontro non abbia ancora entrambi i partecipanti, o che uno dei due salti prima dell'inizio dell'incontro.

La stessa motivazione vale per i coach, in più un atleta può gareggiare anche da solo, senza una persona all'angolo.

## Atleta

Nome Campo	Descrizione	Tipo	Lunghezza	Vincoli
Numero Tessera	Identificativo univoco di un atleta, viene data dalla	Varchar	150	Primary Key, unique, not null
Nome	Nome dell'atleta	Varchar	150	not null
Cognome	Cognome dell'atleta	Varchar	150	not null
Password	Password dell'utente (già hashata)	Varchar	32	not null
Email	Email associata all'utente	Varchar	150	not null, unique
Palestra	Palestra di appartenenza dell'atleta	Varchar	150	Not null
Cintura	Colore della cintura dell'atleta	Enum	150	Not null, valori ammessi: [('W', 'Bianca'), ('G', 'Gialla'), ('B', 'Blu'), ('M', 'Marrone'), ('N', 'Nera'),]
Categoria	Categoria di peso dell'atleta	Varchar	150	not null
storico	Storico incontri atleta	Foreign Key		null, fa riferimento alla tabella Incontro

Il campo "cintura" è stato reso di tipo Enum per consentire al sistema di registrare solo determinati gradi di cintura, per rispettare la traccia.

Il campo "storico" è stato reso una foreign key per semplificare la ricerca dei vari incontri dell'atleta.

Il campo "password" è stato limitato a 32 caratteri per questioni di spazio.

La categoria di peso non è stata resa un campo Enum per facilitare la gestione di categorie di peso "non convenzionali", il controllo sulla validità delle categorie verrà effettuato dall'applicazione stessa.

## Torneo

Nome Campo	Descrizione	Tipo	Lunghezza	Vincoli
ID	Identificativo univoco del record	Intero, Autoincrement		Primary Key, unique, not null
grado minimo	grado minimo richiesto per partecipare al torneo	enum		not null, valori accettati: ('AM', 'Amatore'), ('SP', 'Semi-Pro'), ('PR', 'Professionista')]
Luogo	Luogo dell'evento	Varchar	150	not null
Data	Data dell'evento	Date		not null
Podio	Primi tre classificati	JSON		not null
Disciplina	Disciplina praticata nel torneo	enum		not null, valori ammessi: [('MMA', 'Mixed Martial Arts'), ('K1', 'KickBoxing'), ('BJJ', 'Brazilian Jiu-Jitsu')]
A pagamento	Indica se la partecipazione è gratuita o meno	Bool	1	not null, default = False
id_incontro	Chiave esterna per collegare la tabella agli incontri	Foreign Key		not null, fa riferimento alla tabella Incontro

## Organizzatore

Nome Campo	Descrizione	Tipo	Lunghezza	Vincoli
Numero tessera	Identificativo del record	Varchar	150	Primary Key, not null, unique
Nome	nome dell'organizzatore	varchar	150	not null

Nome Campo	Descrizione	Tipo	Lunghezza	Vincoli
cognome	cognome dell'organizzatore	varchar	150	not null
password	password dell'utente (già hashata)	varchar	32	not null
federazione	ente sotto cui è registrato l'organizzatore	varchar	150	not null
email	email associata all'organizzatore	varchar	150	not null

## Ticket

Nome Campo	Descrizione	Tipo	Lunghezza	Vincoli
ID_ticket	Identificativo unico	Intero, autoincrement		Primary Key, unique, not null
Costo	Prezzo da pagare in sede	Float		not null
data_prenotazione	data dell'evento per cui è stato prenotato il ticket	Date		not null
scadenza	Data validità massima del ticket	DateTime		not null

## Arbitro

Nome Colonna	Descrizione	Tipo	Lunghezza	Vincoli
numTessera	Numero di tessera univoco	Integer		Primary Key
nome	Nome dell'arbitro	Varchar	150	Not null
cognome	Cognome dell'arbitro	Varchar	150	not null
password	Password	Varchar	32	not null
email	Email dell'arbitro	Varchar		not null
età	Età dell'arbitro	Integer		not null
id_incontro	Incontro associato	ForeignKey		Riferimento alla tabella Incontro, not null



## Coach

Nome Colonna	Descrizione	Tipo	Lunghezza	Vincoli
numTessera	Numero di tessera univoco	Intero		Primary Key, unique, not null
nome	Nome del coach	CharField	150	not null
cognome	Cognome del coach	CharField	150	not null
password	Password	CharField	32	not null
email	Email del coach	CharField	150	not null
palestra	Nome della palestra	CharField	150	not null
grado	Grado del coach	CharField	150	not null, valori consentiti [('1', '1 dan'), ('2', '2 dan'), ('3', '3 dan'), ('4', '4 dan'), ('5', '5 dan')]
numTessera_atleta	Atleta associato	ForeignKey	N/A	Riferimento alla tabella Atleta, not null

## Organizzazione

Nome Colonna	Descrizione	Tipo	Lunghezza	Vincoli
numTessera	Organizzatore associato	ForeignKey		Riferimento alla tabella Organizzatore
id_torneo	Torneo associato	ForeignKey		Riferimento alla tabella Torneo

## Vincoli Interrelazionali

Un atleta deve necessariamente avere un record in tabella per poter scaricare ticket, accedere all'area personale per visualizzare/modificare i dati e per farsi iscrivere ad un torneo.

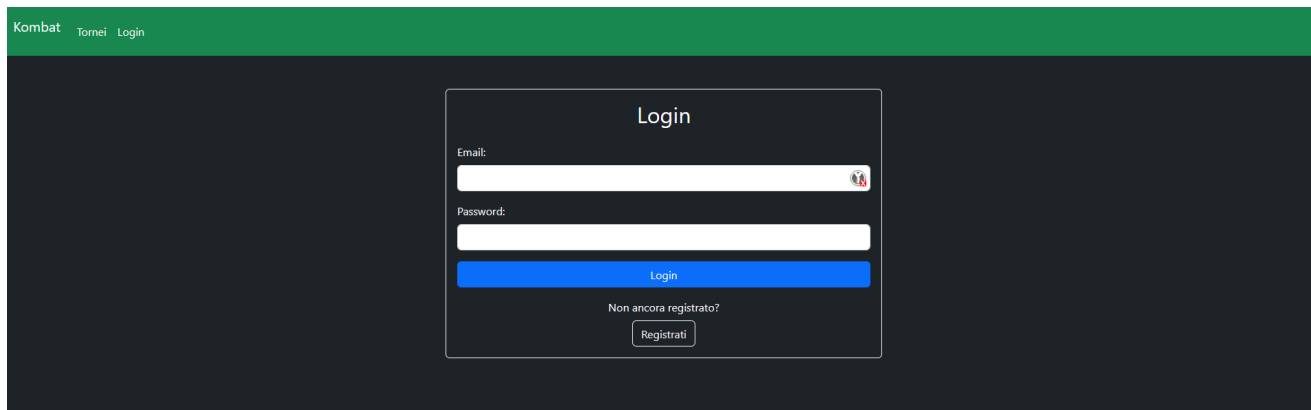
Inoltre, un torneo non può esistere senza un valido organizzatore e un incontro non può verificarsi senza un arbitro valido.

Ad un incontro già organizzato non è permesso non avere i due atleti combattenti. Per fronteggiare l'evenienza di un ritiro da parte di entrambi, l'incontro viene cancellato.

## Descrizione App

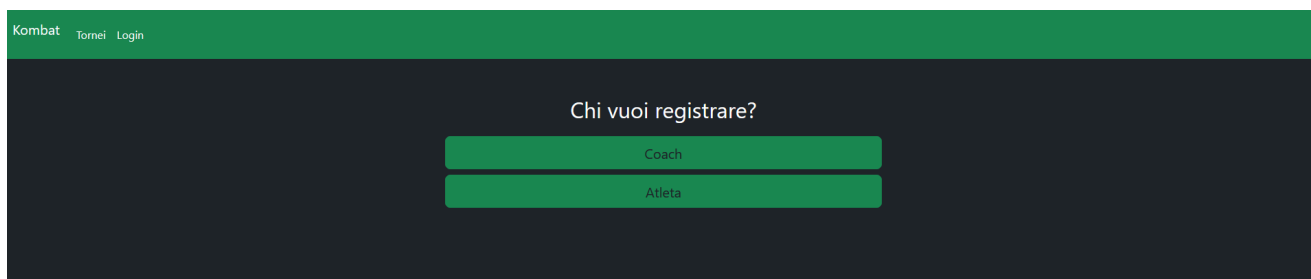
L'applicazione offre varie funzionalità a seconda della tipologia dell'utente loggato.

La prima pagina consultabile è quella del login che si presenta in questo modo:



The screenshot shows the login page of the Kombat app. At the top, there is a green navigation bar with the text "Kombat", "Tornei", and "Login". The main content area is dark gray. In the center, there is a white box with the title "Login". Inside this box, there are two input fields: "Email:" and "Password:". Below the password field is a blue button labeled "Login". Underneath the "Login" button, there is a link that says "Non ancora registrato?" and a button labeled "Registrati".


È anche possibile registrarsi cliccando sul bottone apposito. Il sistema chiederà all'utente di scegliere con quale ruolo registrarsi.



The screenshot shows the registration role selection page of the Kombat app. At the top, there is a green navigation bar with the text "Kombat", "Tornei", and "Login". The main content area is dark gray. In the center, there is a white box with the title "Chi vuoi registrare?". Below the title, there are two buttons: "Coach" and "Atleta".

A seconda della tipologia di utente si potranno inserire i dati corrispondenti:

### Registrazione Atleta



The screenshot shows the athlete registration form of the Kombat app. At the top, there is a green navigation bar with the text "Kombat", "Tornei", and "Login". The main content area is dark gray. The form consists of several input fields: "Email", "Password", "Nome", "Cognome", "Numero di tessera", "Cintura", and "Categoria di peso". Each input field has a placeholder text: "Email", "Password", "Nome", "Cognome", "Numero di tessera", "Seleziona la tua cintura", and "Seleziona la categoria di peso". At the bottom of the form, there is a blue button labeled "Invia".

### Registrazione Coach

Kombat Tornei Login

Email

Password

Nome

Cognome

Numero di tessera

Grado

Seleziona il tuo grado

Invia

Una volta eseguito il login, l'utente visualizzerà la pagina Home personalizzata.

## Homepage Atleta

Kombat Tornei Login

Benvenuto, Luca Bianchi

Storico Incontri

July 15, 2023

Sconfitta

Visualizza

Area Personale

Gli atleti possono visualizzare il proprio storico incontri in questa pagina, e poi possono accedere alla loro area personale

## Homepage Coach

Kombat Tornei Login

Benvenuto, Coach Marroni

Iscrivi Atleta

I coach possono scegliere di iscrivere i loro atleti alle gare (funzionalità non implementata).

## Homepage Arbitro

Kombat Tornei Login

Benvenuto, Arbitro Bianchi

## Homepage Organizzatore

Kombat Tornei Login

Benvenuto, Giorgio Neri

Crea torneo

Gli organizzatori possono creare nuovi tornei (funzione non ancora implementata)

## Area Personale

L'area personale visualizzerà i dati relativi ad ogni utente e permetterà loro di modificarli (eccetto quelli che per legge non possono essere modificati).

### Esempio area personale Atleta

Kombat Tornei Login

#### Dati Personali

Nome  
Luca

Cognome  
Bianchi

Password  
.....

Email address  
luca.bianchi@example.com

Numero di Tessera Federazione  
A12345

Palestra  
Palestra Roma

Cintura  
N

Categoria di peso  
65 Kg

Salva

(Si noti come, in questo caso, il numero di tessera non possa essere modificato)

## Area Tornei

È possibile, inoltre, per tutti gli utenti (anche non registrati) visualizzare tutti i dati dei tornei in tabella.

## Simulazione SQL Injection

Utilizzando la Query API di Django, il sistema è molto resistente ad attacchi di questo tipo, ma ipotizzando di utilizzare una query che utilizza degli input utente non sanitizzati, come ad esempio:

```
SELECT * FROM Atleta WHERE email = '{email}' AND password = '{password}'
```

La query seleziona tutti i dati del record in cui email e password corrispondono all'input inserito dall'utente. Ipotizziamo adesso che il software non sanitizzi l'input ricevuto e non esegua l'hashing della password prima di eseguire la query, un utente malevolo potrebbe inserire un payload del genere: ' OR 1=1 -- nel campo password, andando a modificare la query precedente in:

```
SELECT * FROM Atleta WHERE email = '{email}' AND password = '' OR 1=1 --
```

Questa query ritornerà sempre un valore True, permettendoci di accedere all'account con quella specifica email senza conoscerne la password.

Ci sono diversi modi per difenderci da questo tipo di attacco:

1. Utilizzare l'Object-relational Mapping (ORM) di Django, che astrae l'accesso al database

2. Sanitizzare l'input utente: si può impedire all'utente di inserire caratteri pericolosi (come gli apici)