**Progetto: Sistema di Gestione del Trasporto Pubblico**

Creare un sistema di gestione del trasporto pubblico utilizzando Django. Il sistema deve essere in grado di gestire le linee di autobus, le fermate, gli autobus, i conducenti e i passeggeri, implementando le funzionalità richieste.

1. Configurazione del Progetto

- Creare un progetto Django e una o più applicazioni.

- Configurare il progetto per utilizzare SQLite come database.

2. Modelli Django

- **Line**

- `line\_number: str`

- `name: str`

- `stops: ManyToManyField(Stop)`

- **Stop**

- `stop\_id: str`

- `name: str`

- `latitude: float`

- `longitude: float`

- `lines: ManyToManyField(Line)`

- **Bus**

- `bus\_id: str`

- `line: ForeignKey(Line)`

- `capacity: int`

- `passengers: ManyToManyField(Passenger)`

- **Driver**

- `driver\_id: str`

- `name: str`

- `surname: str`

- `assigned\_bus: ForeignKey(Bus)`

- **Passenger**

- `passenger\_id: str`

- `name: str`

- `surname: str`

3. Viste

- Pagina Principale: Visualizza tutte le linee degli autobus

- Dettaglio Linea: Mostra i dettagli di una linea specifica.

- Aggiungi Linea: Form per aggiungere una nuova linea.

- Rimuovi Linea: Funzionalità per rimuovere una linea.

- Lista delle Fermate: Visualizza una lista di tutte le fermate.

- Dettaglio Fermata: Mostra i dettagli di una fermata specifica.

- Lista degli Autobus: Visualizza una lista di tutti gli autobus.

- Dettaglio Autobus: Mostra i dettagli di un autobus specifico.

- Aggiungi Autobus: Form per aggiungere un nuovo autobus.

- Rimuovi Autobus: Funzionalità per rimuovere un autobus.

- Lista dei Conducenti: Visualizza una lista di tutti i conducenti.

- Dettaglio Conducente: Mostra i dettagli di un conducente specifico.

- Assegna Conducente: Form per assegnare un conducente a un autobus.

- Rimuovi Conducente: Funzionalità per rimuovere un conducente.

- Lista dei Passeggeri: Visualizza una lista di tutti i passeggeri.

- Dettaglio Passeggero: Mostra i dettagli di un passeggero specifico.

- Salita Passeggero: Funzionalità per far salire un passeggero su un autobus.

- Discesa Passeggero: Funzionalità per far scendere un passeggero dall'autobus.

4. Form

- Creare form Django per aggiungere e rimuovere autobus.

- Creare form per assegnare e rimuovere conducenti.

- Creare form per far salire e scendere passeggeri dagli autobus.

5. URL

- Definire i pattern URL per ciascuna vista.

- Assicurarsi che i pattern URL siano chiari e intuitivi.

6. File Statici

- Includere CSS per stilizzare i template HTML.

- Eventualmente includere JavaScript per migliorare l'interattività.

7. Testing

- Scrivere test unitari per modelli, viste e form.

- Assicurarsi che tutte le funzionalità funzionino come previsto.

8. Documentazione

- Fornire una documentazione chiara su come configurare il progetto.

- Includere istruzioni per eseguire il server di sviluppo.

- Documentare lo scopo e l'uso di ciascun componente principale (modelli, viste, template).

**Sfide Aggiuntive (Opzionali)**

- Funzionalità di Ricerca: Implementare una funzionalità di ricerca per trovare linee, fermate, autobus, conducenti e passeggeri.

- Monitoraggio della Capacità: Implementare una funzionalità per monitorare la capacità degli autobus e segnalare quando sono pieni.

- Pagina di Statistiche: Creare una pagina che mostri statistiche come il numero di linee, fermate, autobus, conducenti e passeggeri.

- Design Responsivo: Assicurarsi che il sito web sia mobile-friendly.

- Caricamento Massivo: Implementare una funzionalità per caricare linee, fermate o passeggeri in massa da un file CSV.

**Requisiti di Consegna**

- Codice sorgente caricato in un repository Git (ad es., GitHub, GitLab).

- Un file README con istruzioni per la configurazione.

- Screenshot o un breve video che dimostri il progetto funzionante.

**Criteri di Valutazione**

- Funzionalità: Il sistema soddisfa tutti i requisiti specificati.

- Qualità del Codice: Il codice è ben organizzato, leggibile e segue le migliori pratiche di Django.

- Interfaccia Utente: L'interfaccia utente è user-friendly e esteticamente gradevole.

- Testing: Test adeguati per assicurare che tutte le funzionalità funzionino correttamente.

- Documentazione