

"Orizon - API RESTful per Viaggi Responsabili"

GitHub: [Github](#)
Havolli Erjon



Overview

- ▶ Introduzione 01
- ▶ Obiettivi del Progetto 02
- ▶ Scelte Tecnologiche 03
- ▶ Architettura del Progetto 04
- ▶ Dettagli delle API 05
- ▶ Testing 06
- ▶ Conclusioni 07





Introduzione

Descrizione del progetto:

- Breve descrizione del progetto Orizon e la sua missione: promuovere viaggi responsabili e sostenibili.
- Obiettivo: Creare un backend con API RESTful per gestire utenti, prodotti e ordini.

Obiettivi del Progetto

- 01 Creare API RESTful per la gestione di utenti, prodotti e ordini.
- 02 Implementare operazioni CRUD (Create, Read, Update, Delete) per utenti, prodotti e ordini.
- 03 Gestire la connessione a MongoDB per archiviare i dati.
- 04 Fornire un'architettura chiara e scalabile.

Scelte Tecnologiche

Node.js:

Motivo della scelta:
"Node.js è stato
scelto per la sua
velocità e
l'efficienza
nell'elaborazione
delle richieste,
particolarmente
adatto per
un'applicazione
basata su API."

Express.js:

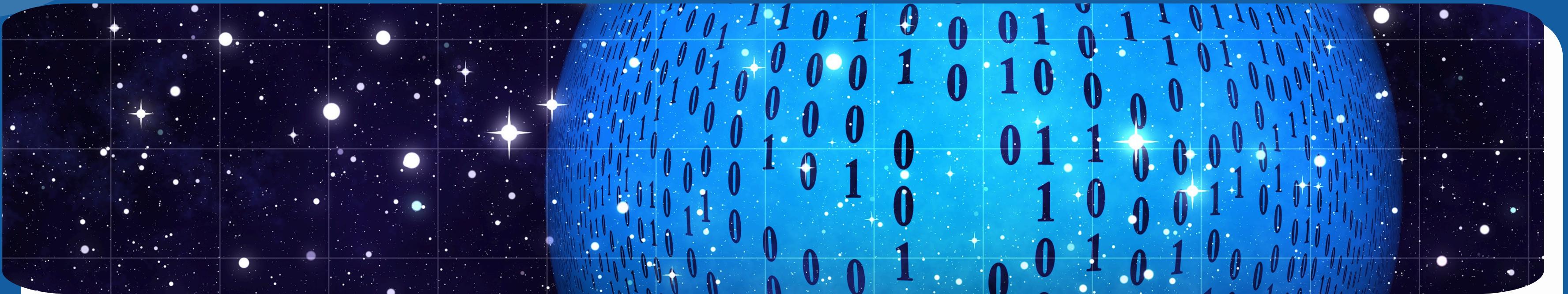
Motivo della scelta:
"Express è un
framework
minimale e
flessibile per
Node.js, che
semplifica la
gestione delle rotte
e delle richieste
HTTP."

MongoDB:

Motivo della scelta:
"MongoDB è stato
scelto per la sua
natura NoSQL, che
permette di gestire
facilmente dati
non strutturati e di
scalare
rapidamente
l'applicazione."



Architettura del Progetto



Struttura delle cartelle:

- /models: contiene gli schemi di MongoDB per utenti, prodotti e ordini.
- /controllers: gestisce la logica di business per ciascuna entità.
- /routes: definisce le rotte dell'API per ciascuna entità.
- /index.js: avvia l'applicazione e gestisce la connessione al database.

Dettagli delle API

API Utenti:

- CRUD per gestire gli utenti: creazione, lettura, aggiornamento e cancellazione.
- Esempio di endpoint: POST /api/users/, GET /api/users/, PUT /api/users/:userID, DELETE /api/users/:userID.

API Prodotti:

- CRUD per i prodotti: creazione, lettura, aggiornamento e cancellazione.
- Esempio di endpoint: POST /api/products/, GET /api/products/, PUT /api/products/:productId, DELETE /api/products/:productId.

API Ordini:

- CRUD per gli ordini: creazione, lettura, aggiornamento e cancellazione.
- Esempio di endpoint: POST /api/orders/, GET /api/orders/, PUT /api/orders/:orderId, DELETE /api/orders/:orderId.

Testing

- Le API sono state testate manualmente tramite Postman, simulando scenari reali:
 - Creazione utente
 - Inserimento prodotto
 - Creazione ordine con più prodotti e utenti
 - Modifica e cancellazione dei dati
- Tutti i test hanno verificato:
 - La corretta risposta del server (status code 200, 201, 404, ecc.)
 - La validità del JSON restituito
 - La coerenza dei dati nel database

Sicurezza

- MongoDB è stato scelto come database.

Misure per prevenire NoSQL Injection:

- Validazione dei dati in input tramite schema Mongoose
- Sanificazione dei parametri nelle richieste
- Evitato l'uso di operatori pericolosi come \$where o query dinamiche non controllate



CONCLUSIONI

"Il progetto Orizon API è completo e pronto per la produzione. Le API sono scalabili, sicure e facilmente manutenibili."