## **Introduzione al progetto**

Il progetto di creazione di un sito scolastico è stato realizzato utilizzando Angular per la parte frontend, Spring Boot per la parte back end e PostgreSQL come database. L'obiettivo del progetto è stato quello di creare un sito web per una scuola in cui i visitatori possano visualizzare le informazioni della scuola e degli studenti, e dove gli utenti registrati possano gestire il proprio profilo e visualizzare i propri voti e i propri compiti.

Per la parte frontend, Angular è stato scelto per la sua facilità di utilizzo e per le numerose funzionalità che offre, come l'utilizzo dei componenti, la gestione delle rotte e la comunicazione con il back end tramite le chiamate HTTP.

Per la parte back end, invece, Spring Boot è stato scelto per la sua capacità di creare rapidamente un'applicazione web robusta e scalabile, utilizzando molte delle funzionalità offerte da Spring, come l'inversione di controllo e l'accesso ai dati tramite Spring Data.

Infine, per la gestione dei dati, è stato scelto PostgreSQL, un database relazionale open source, per la sua affidabilità, la scalabilità e le numerose funzionalità di sicurezza che offre.

Il progetto comprende anche funzionalità come la gestione degli utenti, la registrazione, l'autenticazione e l'autorizzazione, la gestione dei voti e dei compiti degli studenti e la visualizzazione delle informazioni della scuola. Il progetto è stato realizzato con un approccio modulare, suddividendo il frontend e il backend in moduli separati, per rendere più facile la manutenzione e l'aggiornamento del codice.

## **Tecnologie utilizzate**

## **Spring boot**

### **Descrizione di Spring Boot e come è stato utilizzato nel progetto**

Spring Boot è un framework per lo sviluppo di applicazioni web che si basa sul framework Spring. Si tratta di un progetto open source che permette di creare facilmente applicazioni stand-alone, in quanto include al suo interno un server web embedded, senza la necessità di configurazioni particolari.

Nel progetto in questione, Spring Boot è stato utilizzato come principale framework per la creazione di un API RESTful. In particolare, è stato scelto per la sua flessibilità e per la facilità con cui si possono configurare i servizi web e le componenti della web application.

### **Descrizione delle dipendenze utilizzate nel progetto**

* Spring-boot-starter-data-jpa: è una dipendenza di Spring Boot che include il supporto per JPA (Java Persistence API). JPA è uno standard per l'interfacciamento con i database relazionali in Java, e fornisce un'interfaccia orientata agli oggetti ai dati nel database. Questa dipendenza include le dipendenze per Hibernate, un framework per l'ORM (Object-Relational Mapping), e per un provider di connessione al database.
* Spring-boot-starter-web: è una dipendenza di Spring Boot che include le dipendenze necessarie per creare un'applicazione web, come ad esempio il framework Spring MVC (Model-View-Controller) per la gestione delle richieste HTTP.
* Spring-boot-devtools: è una dipendenza di Spring Boot che fornisce una serie di strumenti utili per lo sviluppo, come ad esempio il reloading automatico dell'applicazione durante lo sviluppo.
* Postgresql: è una dipendenza che fornisce il driver JDBC per il database PostgreSQL. Il driver JDBC è un'implementazione di un'interfaccia standard che permette a Java di interagire con il database.
* Lombok: è una dipendenza che fornisce una serie di strumenti per semplificare la scrittura di codice Java, come ad esempio la generazione automatica di getter e setter per le proprietà delle classi.
* Jjwt-api: è una dipendenza per la gestione di JSON Web Token (JWT), un meccanismo per l'autenticazione e l'autorizzazione sicura e decentralizzata. Questa dipendenza include solo l'API per la gestione di JWT, non l'implementazione concreta.
* Jjwt-impl: è una dipendenza che fornisce l'implementazione concreta della gestione di JWT.
* Jjwt-jackson: è una dipendenza che fornisce la serializzazione e deserializzazione di JWT in formato JSON, usando il framework Jackson;
* Spring-boot-starter-mail: è una dipendenza di Spring Boot che include il supporto per l'invio di email tramite JavaMail.
* Springfox-swagger2: è una dipendenza per la documentazione delle API tramite il framework Swagger. Swagger permette di creare una documentazione automatica delle API esposte dall'applicazione, rendendo più facile la loro comprensione e utilizzo.
* Springfox-boot-starter: è una dipendenza che include le dipendenze necessarie per l'utilizzo di Swagger in un'applicazione Spring Boot.
* Springfox-swagger-ui: è una dipendenza che include la UI per la visualizzazione della documentazione delle API generata da Swagger.
* Spring-boot-starter-security: è una dipendenza di Spring Boot che include le dipendenze necessarie per la gestione della sicurezza dell'applicazione, come ad esempio l'autenticazione e l'autorizzazione degli utenti.

Queste sono solo alcune delle dipendenze utilizzate nel progetto, ma coprono le principali funzionalità utilizzate.

### C**onfigurazione di Spring Boot per il progetto**

La configurazione del progetto si è instaurata nelle properties qui sopra indicate:

Dove:

* server.port: definisce la porta sulla quale il server Spring Boot deve essere esposto.
* spring.datasource.url: definisce l'URL per accedere al database.
* spring.datasource.username: definisce l'username utilizzato per accedere al database.
* spring.datasource.password: definisce la password utilizzata per accedere al database.
* spring.jpa.hibernate.ddl-auto: definisce come Hibernate deve gestire la creazione del database. In questo caso, con "update", Hibernate crea il database se non esiste e aggiorna lo schema se è cambiato.
* spring.jpa.show-sql: se impostato su "true", Spring Boot mostra le query SQL generate da Hibernate sulla console.
* spring.jpa.properties.hibernate.dialect: definisce il dialetto di Hibernate per la gestione delle query SQL. In questo caso, viene utilizzato il dialetto specifico per PostgreSQL.
* spring.jpa.properties.hibernate.format\_sql: se impostato su "true", Hibernate formatta le query SQL generate per una migliore leggibilità.
* spring.mail.host: definisce l'host del server di posta utilizzato per inviare le email.
* spring.mail.port: definisce la porta utilizzata per l'invio delle email.
* spring.mail.username: definisce l'username utilizzato per accedere al server di posta.
* spring.mail.password: definisce la password utilizzata per accedere al server di posta.
* spring.mail.properties.mail.smtp.auth: se impostato su "true", abilita l'autenticazione per l'invio delle email.
* spring.mail.properties.mail.smtp.starttls.enable: se impostato su "true", abilita il protocollo STARTTLS per l'invio delle email.

## **Angular**

### **Descrizione di Angular e come è stato utilizzato nel progetto**

L'utilizzo di Angular per la creazione di un sito scolastico ha portato numerosi vantaggi. Innanzitutto, la possibilità di creare una struttura modulare e scalabile, in cui ogni sezione del sito può essere sviluppata come un componente Angular. Inoltre, grazie al sistema di routing offerto da Angular, è stato possibile creare una navigazione fluida e veloce tra le diverse sezioni del sito.

Un altro vantaggio è la facilità di gestione dei dati, grazie alla possibilità di utilizzare servizi che si interfacciano con i server di backend per recuperare le informazioni necessarie. Inoltre, il sistema di binding di Angular ha permesso di creare un’ interfaccia utente dinamica e reattiva, in grado di interagire con l'utente in tempo reale.

Infine, l'utilizzo di Angular ha permesso di utilizzare numerosi plugin e librerie, offrendo un'ampia gamma di strumenti per creare un sito web di alta qualità e performante. In sintesi, l'utilizzo di Angular è stato fondamentale per la creazione di un sito scolastico moderno, modulare e interattivo, in grado di offrire una piacevole esperienza utente.

### **Descrizione di Bootstrap e principali funzionalità**

Bootstrap è una libreria CSS e JavaScript gratuita e open source, sviluppata da Twitter, che offre una serie di strumenti e componenti per la creazione di siti web responsivi e moderni. È stato progettato per semplificare il processo di sviluppo di un sito web, fornendo una vasta gamma di funzionalità, come ad esempio la griglia di layout, i componenti predefiniti, i tipi di carattere, gli stili di pulsanti e le animazioni.

Una delle principali funzionalità di Bootstrap è la griglia di layout, che permette di creare una struttura a colonne per organizzare i contenuti della pagina in modo flessibile e adattabile a diversi dispositivi. La griglia di Bootstrap è basata su un sistema a 12 colonne, che permette di suddividere lo spazio della pagina in parti uguali o in proporzione diversa, a seconda delle esigenze.

Un'altra funzionalità molto utile di Bootstrap sono i componenti predefiniti, come le barre di navigazione, le schede, le liste di elementi, le caselle di avviso e le finestre modal. Bootstrap offre anche una vasta gamma di tipi di carattere, che possono essere facilmente utilizzati all'interno del progetto, e una serie di stili di pulsanti, che consentono di creare pulsanti personalizzati con diversi colori e dimensioni.

Infine, Bootstrap fornisce anche un set di animazioni e transizioni, che possono essere utilizzati per rendere la navigazione del sito più fluida e piacevole per l'utente.

Nella creazione di un sito scolastico, Bootstrap può essere utilizzato per creare un design moderno e responsive, che si adatta alle esigenze dei diversi dispositivi, come smartphone, tablet e computer. La griglia di Bootstrap può essere utilizzata per organizzare i contenuti in modo chiaro e intuitivo e inoltre, Bootstrap può essere personalizzato per adattarsi al branding della scuola, utilizzando i colori e i tipi di carattere appropriati.

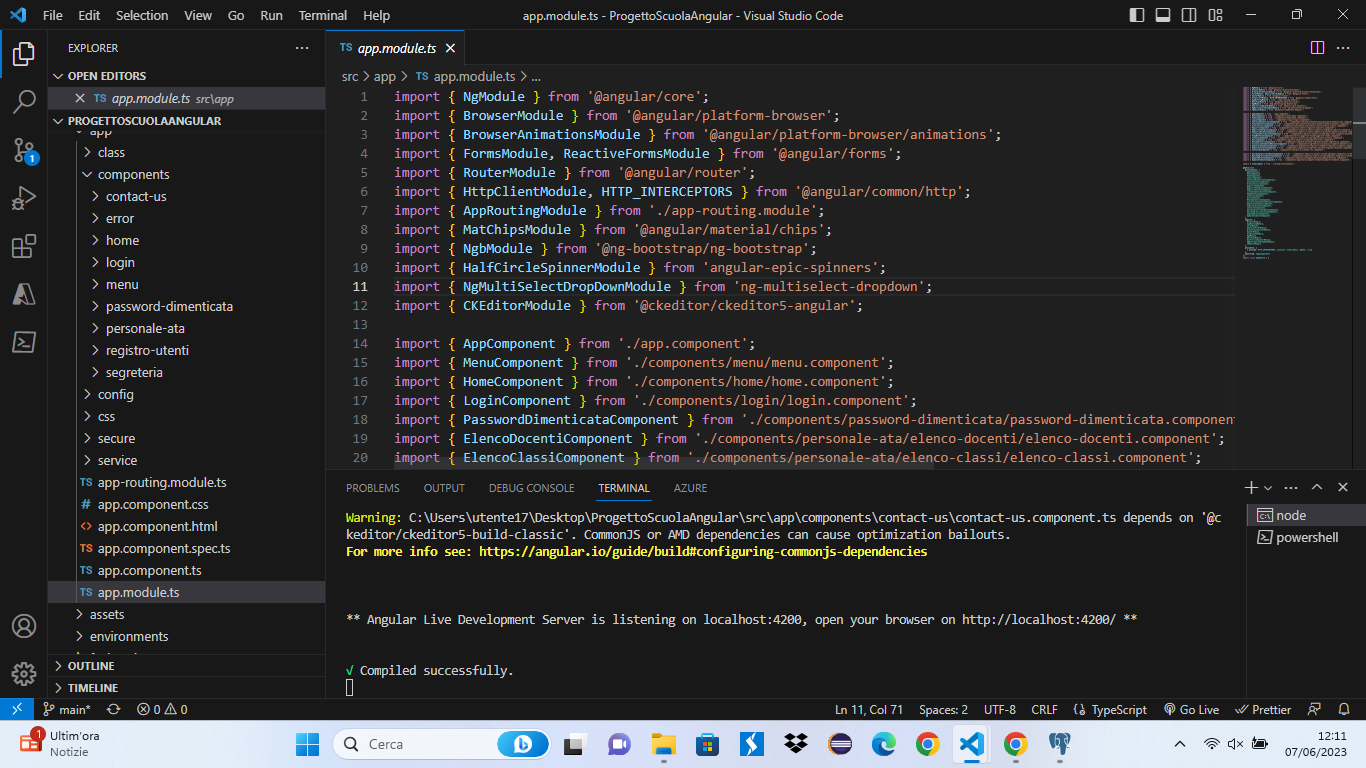
### 

### 

### 

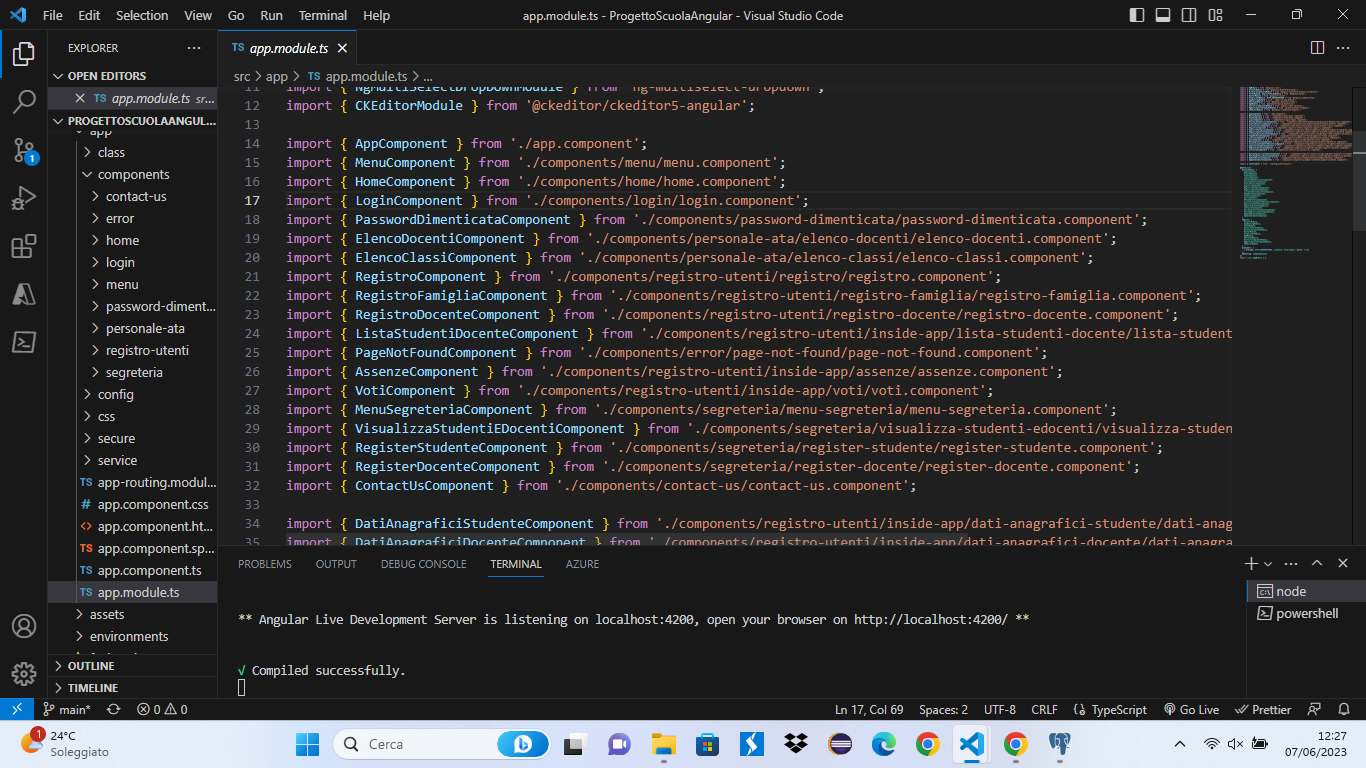
### Descrizione dei moduli utilizzati nel progetto

La dichiarazione dei moduli per poi essere utilizzati nel progetto sono rappresentati nell'immagine qui sottoposta:



* BrowserModule**:** È un modulo fondamentale per l'applicazione Angular che viene importato dal modulo principale (AppModule) per consentire l'esecuzione dell'applicazione nel browser. Fornisce funzionalità essenziali come il rendering del modello, la gestione degli eventi e altre funzionalità di base.
* BrowserAnimationsModule: È un modulo che fornisce supporto per le animazioni CSS in Angular. Importandolo nel modulo principale dell'applicazione, permette di utilizzare le animazioni CSS predefinite e personalizzate nelle componenti dell'app.
* FormsModule e ReactiveFormsModule: Sono moduli che forniscono il supporto per la gestione dei moduli dei moduli in Angular. FormsModule è utilizzato per gestire i moduli di moduli con l'approccio di template-driven form, mentre ReactiveFormsModule è utilizzato per gestire i moduli di moduli con l'approccio di reactive form.
* RouterModule: È un modulo di routing in Angular che fornisce strumenti per la gestione delle route dell'applicazione. Consente di definire le route dell'applicazione e di navigare tra le diverse viste o componenti dell'app in base all'URL.
* HttpClientModule: È un modulo che fornisce supporto per l'interazione con i servizi web attraverso le richieste HTTP. Permette di effettuare richieste HTTP come GET, POST, PUT, DELETE, ecc., e gestire le risposte ricevute dai server.
* MatChipsModule: È un modulo del framework di componenti Angular Material che fornisce componenti per la visualizzazione e la gestione dei chips, ovvero elementi di interfaccia utente che rappresentano un'etichetta o un pezzo di contenuto brevi.
* NgbModule: È un modulo del framework Angular Bootstrap (ng-bootstrap) che integra le componenti di Bootstrap nell'applicazione Angular. Fornisce componenti come modali, tooltip, carousel, ecc., che possono essere utilizzate per creare un'interfaccia utente responsive e moderna.
* HalfCircleSpinnerModule: È un modulo di terze parti chiamato "angular-epic-spinners" che fornisce un insieme di componenti di spinner animati. In particolare, il modulo HalfCircleSpinnerModule offre un componente di spinner a forma di mezza circonferenza.
* NgMultiSelectDropDownModule: È un modulo di terze parti chiamato "ng-multiselect-dropdown" che fornisce un componente di selezione multipla a discesa (dropdown) per Angular. Consente di selezionare più opzioni da un elenco predefinito.
* CKEditorModule: È un modulo che integra l'editor di testo ricco CKEditor in un'applicazione Angular. Permette di aggiungere funzionalità di editing di testo avanzate e personalizzate utilizzando CKEditor nell'app.

### Descrizione dei componenti utilizzati nel progetto

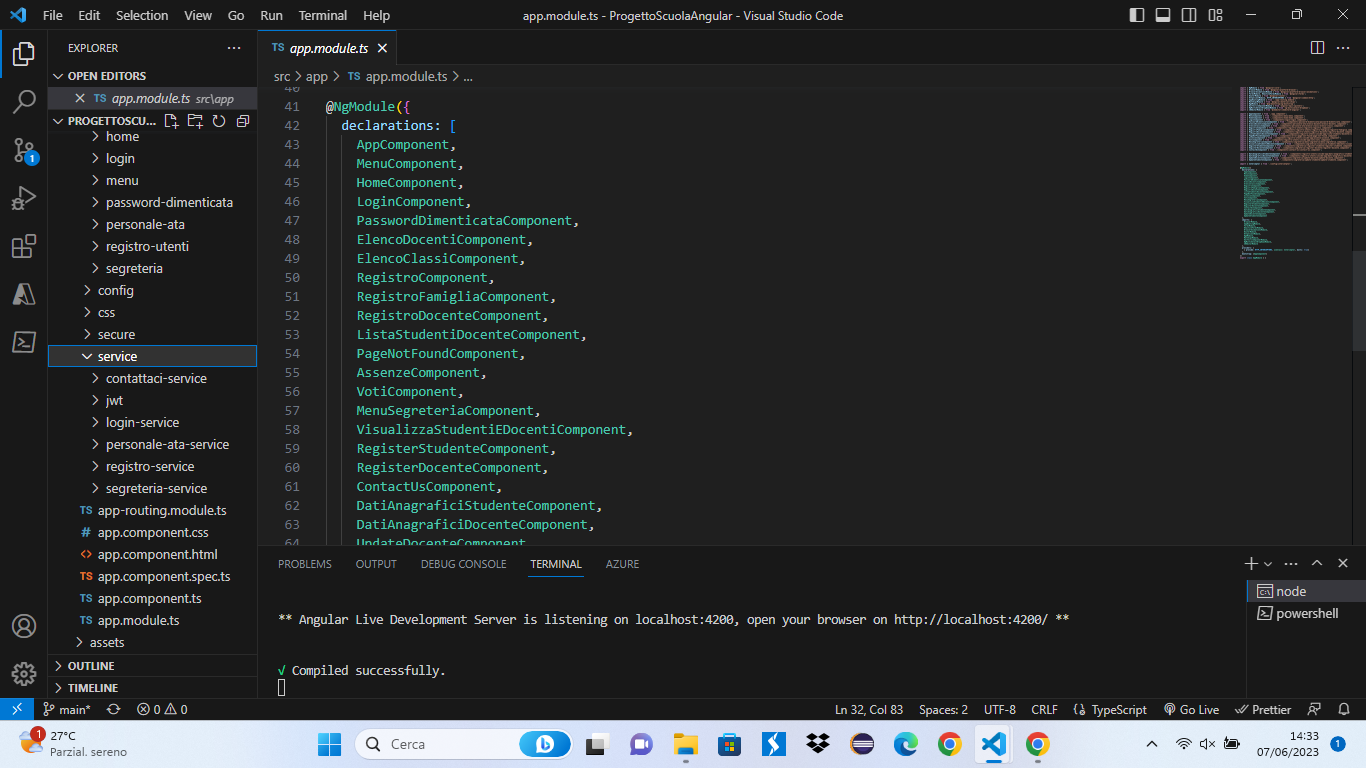
I seguenti componenti saranno utilizzati per permettere l’interazione tra utente e sistema. La foto qui sopra citata indica tutti i componenti mentre la descrizione di ogni singolo si trova di seguito:

* AppComponent: Questo è il componente radice dell'applicazione. Di solito, contiene il layout principale e coordina gli altri componenti all'interno dell'applicazione.
* MenuComponent: Questo componente rappresenta il menu di navigazione dell'applicazione e fornisce opzioni per spostarsi tra le diverse sezioni o pagine.
* HomeComponent: Questo componente rappresenta la pagina principale dell'applicazione.
* LoginComponent: Questo componente gestisce il processo di autenticazione e permette agli utenti di effettuare il login.
* PasswordDimenticataComponent: Questo componente gestisce il processo di recupero della password nel caso in cui un utente l'abbia dimenticata.
* ElencoDocentiComponent: Questo componente visualizza l'elenco dei docenti.
* ElencoClassiComponent: Questo componente visualizza l'elenco delle classi disponibili.
* RegistroComponent: Questo componente gestisce il processo di registrazione di un utente nel sistema.
* RegistroFamigliaComponent: Questo componente gestisce il registro elettronico di una famiglia.
* RegistroDocenteComponent: Questo componente gestisce il registro elettronico di un docente.
* ListaStudentiDocenteComponent: Questo componente visualizza l'elenco degli studenti associati a un docente.
* PageNotFoundComponent: Questo componente viene visualizzato quando l'URL richiesto non corrisponde a nessuna delle rotte definite nell'applicazione.
* AssenzeComponent: Questo componente gestisce la possibilità di visualizzazione e giustifica delle assenze degli studenti.
* VotiComponent: Questo componente gestisce la visualizzazione dei voti degli studenti.
* MenuSegreteriaComponent: Questo componente rappresenta il menu per le funzionalità di segreteria all'interno dell'applicazione.
* VisualizzaStudentiEDocentiComponent: Questo componente permette la visualizzazione, la modifica e anche l'eliminazione sia degli studenti che dei docenti da parte del personale della segreteria.
* RegisterStudenteComponent: Questo componente gestisce il processo di registrazione di uno studente nel sistema da parte del personale di segreteria.
* RegisterDocenteComponent: Questo componente gestisce il processo di registrazione di un docente nel sistema da parte del personale di segreteria.
* ContactUsComponent: Questo componente rappresenta la pagina di contatto o di assistenza dell'applicazione.

### 

### 

### **Descrizione dei servizi utilizzati nel progetto**

* 
* contattaci-service: Service utilizzato nel inviare una chiamata, di tipo HTTP al server,.il quale fine è quello di permettere al server di poter inviare una mail di supporto a nome dell’utente alla azienda o società, in questo caso una scuola.
* jwt: Service utilizzato per poter decodificare il token JWT in un oggetto avente come parametri: lo username dell’utente, la data di creazione del token e la data in cui il token termina.
* login-service: Service utilizzato per la registrazione di un nuovo utente all’interno del database e anche per poter effettuare l’accesso all’utente che si sta loggando.
* personale-ata-service: Service utilizzato per poter ottenere la lista dei docenti e delle classi disponibili nella scuola.
* registro-service: Service utilizzato per poter permettere la visualizzazione dei propri dati anagrafici all’utente, di poter visualizzare le assenze e i voti da parte dello studente. Mentre da parte del docente permette la visualizzazione degli studenti che appartengono a lui e anche la possibilità di poter aggiungere voti e assenze ad uno studente.
* segreteria-service: Service utilizzato per permettere al personale della segreteria di poter visualizzare tutti i docenti e studenti e di poter effettuare una ricerca per poter trovare gli studenti (ancora da implementare). Inoltre questo service permette la modifica e la cancellazione sia dei docenti che degli studenti.

## **Postman**

### **Descrizione di Postman e come è stato utilizzato nel progetto**

Postman è un'applicazione che consente agli sviluppatori di testare e collaborare nello sviluppo di API. Fornisce un'interfaccia utente intuitiva per inviare richieste HTTP alle API, visualizzare le risposte ricevute e analizzare i dati restituiti.

Inoltre postman è stato utilizzato come strumento per testare e debuggare le API. Ecco come è stato utilizzato:

1. Test delle API: Postman è stato utilizzato per inviare richieste HTTP alle API sviluppate nel progetto e verificare le risposte ricevute. Sono state testate diverse richieste come GET, POST, PUT, DELETE per verificare il corretto funzionamento delle API. È possibile impostare i parametri, gli header e il corpo della richiesta direttamente nell'interfaccia di Postman e osservare le risposte restituite dalle API.
2. Debugging delle API: Postman è stato utilizzato per identificare e risolvere eventuali problemi o errori nelle API. Attraverso il test delle richieste, è stato possibile esaminare le risposte ottenute e confrontarle con le aspettative. In caso di errori, è stato possibile analizzare gli errori restituiti dalle API e apportare le necessarie correzioni.

### **Esempi di richieste API con Postman**

## **Swagger**

### **Descrizione di Swagger e come è stato utilizzato nel progetto**

Swagger è stato utilizzato nel tuo progetto per progettare, testare e condividere le API in modo efficiente. Inoltre l’'utilizzo di Swagger ha migliorato la comprensione delle API, la collaborazione tra i membri del team e ha agevolato l'integrazione delle API da parte del client.

Swagger è stato utilizzato come parte integrante dello sviluppo delle API. Ecco come è stato utilizzato:

1. Progettazione delle API: Swagger ha fornito un ambiente visuale e intuitivo per progettare le API. Ha semplificato la progettazione delle API fornendo suggerimenti, validazioni e strumenti di completamento automatico.

### 

## **Database**

### **Descrizione del database**

Come database si è utilizzato PostgreSQL.

PostgreSQL è un sistema di gestione di database relazionale (RDBMS) open-source che fornisce un'ampia gamma di funzionalità per la gestione dei dati. È noto per la sua affidabilità, scalabilità e capacità di gestire carichi di lavoro complessi. Nel contesto del tuo progetto scolastico, PostgreSQL è stato scelto come database per gestire le informazioni relative alla scuola.

### **Modello di dati utilizzato nel progetto**

### **Descrizione delle tabelle utilizzate nel progetto**

Il database scolastico utilizza PostgreSQL per memorizzare e gestire i dati relativi alle assenze degli studenti, alle classi, ai docenti, alle materie, ai token per il recupero delle password e alle valutazioni. Di seguito, viene fornita una descrizione delle tabelle principali presenti nel database:

1. Assenza: Questa tabella registra le assenze degli studenti.
2. Classi: Questa tabella contiene le informazioni sulle classi presenti nella scuola.
3. Docenti\_Classe\_Table: Questa tabella stabilisce una relazione molti-a-molti tra le classi e i docenti. Contiene informazioni come l'ID della classe e l'ID del docente, consentendo di associare i docenti alle classi che insegnano.
4. Docente\_Materia\_Table: Questa tabella stabilisce una relazione molti-a-molti tra i docenti e le materie. Contiene informazioni come l'ID del docente e l'ID della materia, consentendo di associare i docenti alle materie che insegnano.
5. Docenti: Questa tabella contiene le informazioni sui docenti.
6. Materia: Questa tabella contiene le informazioni sulle materie insegnate nella scuola.
7. Studenti: Questa tabella contiene le informazioni sugli studenti iscritti alla scuola.
8. Token: Questa tabella registra i token per la reimpostazione delle password.
9. Valutazione: Questa tabella registra le valutazioni degli studenti.

### **Descrizione della migrazione tra Oracle e PostgreSQL**

La migrazione è stata effettuata per testare la versatilità del codice e quindi il suo funzionamento anche cambiando database.

La migrazione è avvenuta da postgreSQL a Oracle e per effettuare questa migrazione si è utilizzato SQLIE, il quale non è nient'altro che uno strumento gratuito per esportare e importare i dati da un database ad un altro, semplicemente mettendo il server name che nel nostro caso era ‘localhost’ e inserire i dati di accesso del database da cui ricavare i dati.

Successivamente i dati verranno esportati in un file che potrà essere utilizzato per l’import, sempre grazie a questo strumento.