



TEDx

4all

Compito 4: Flutter

Pesenti Luca (mat. 1079602)

Pesenti Alessandro (mat. 1082457)



Tecnologie Cloud e Mobile 2024





INDICE

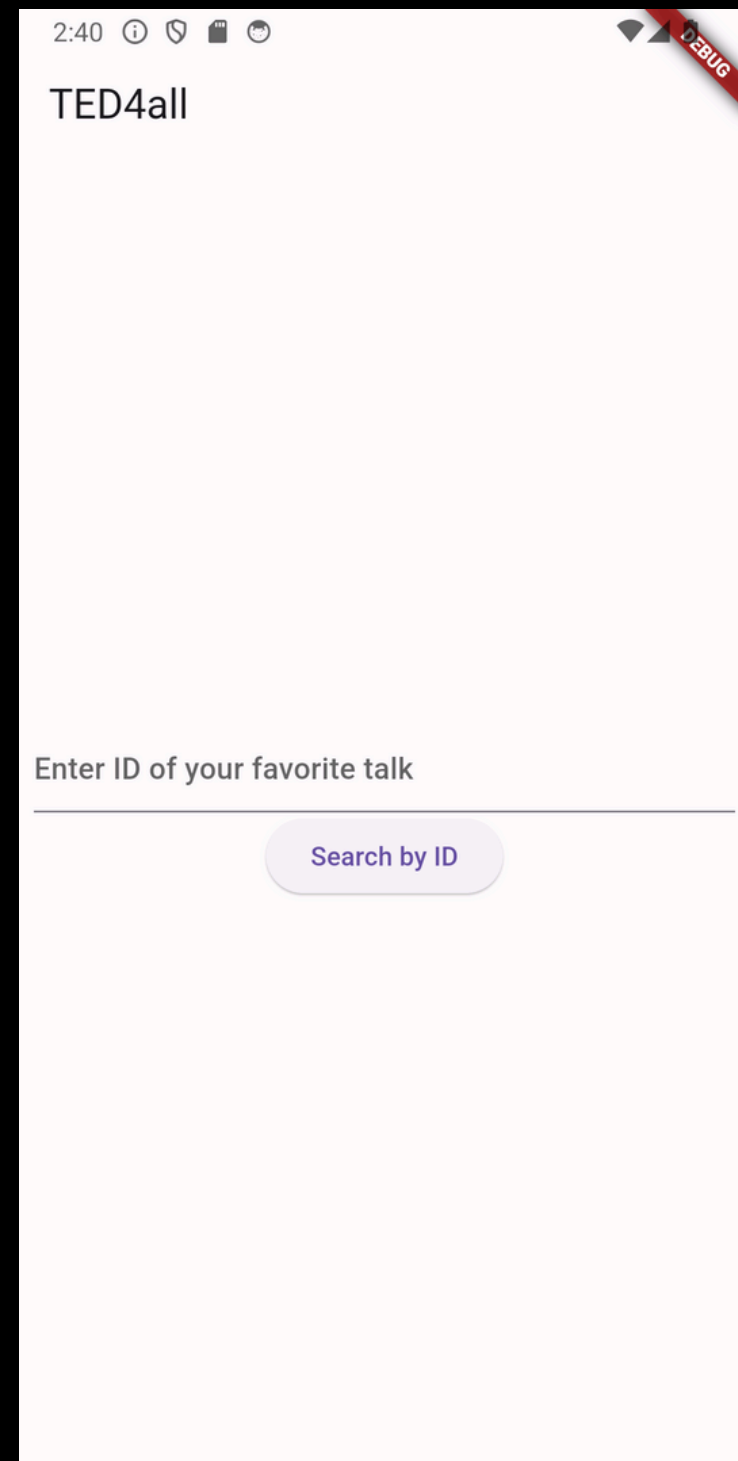
1 Watch Next

2 Transcript

3 Criticità e sviluppi futuri

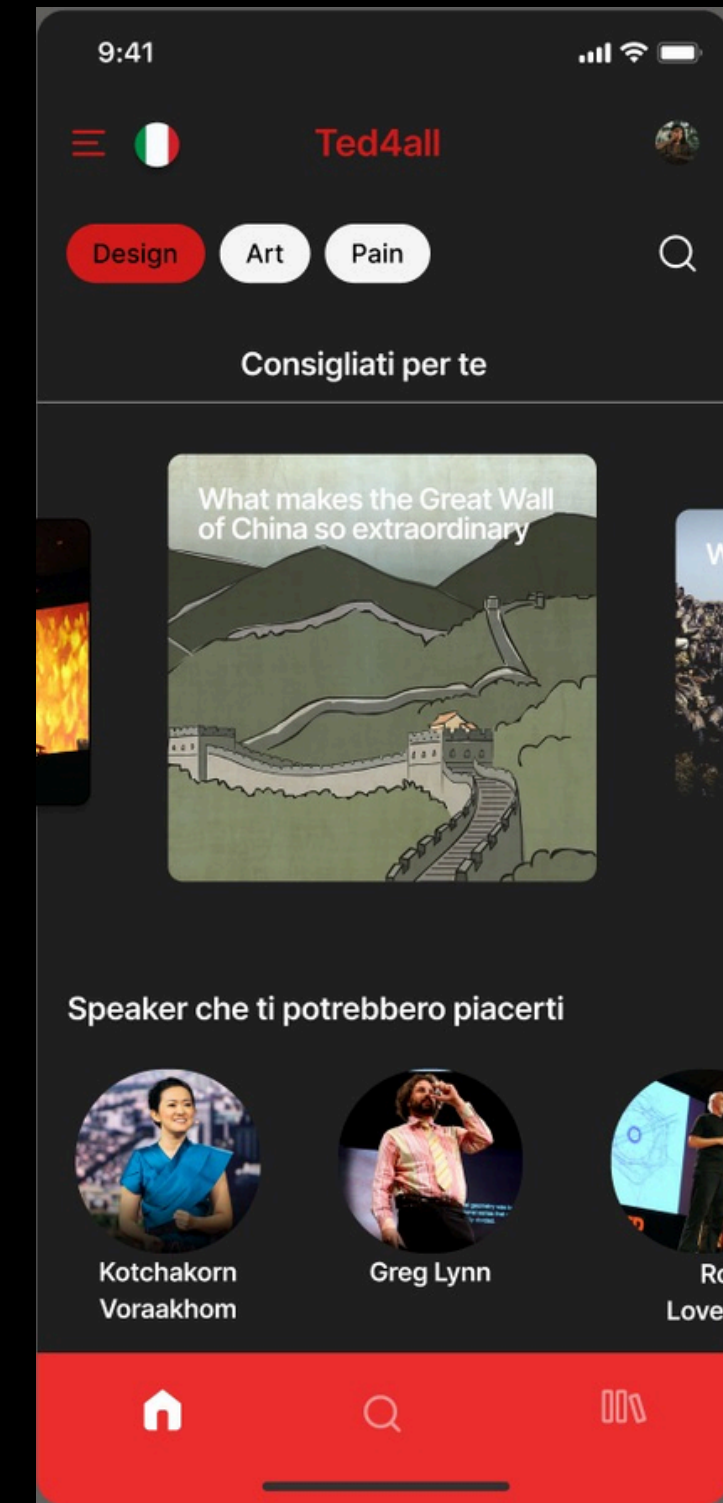


PAGINA PRINCIPALE

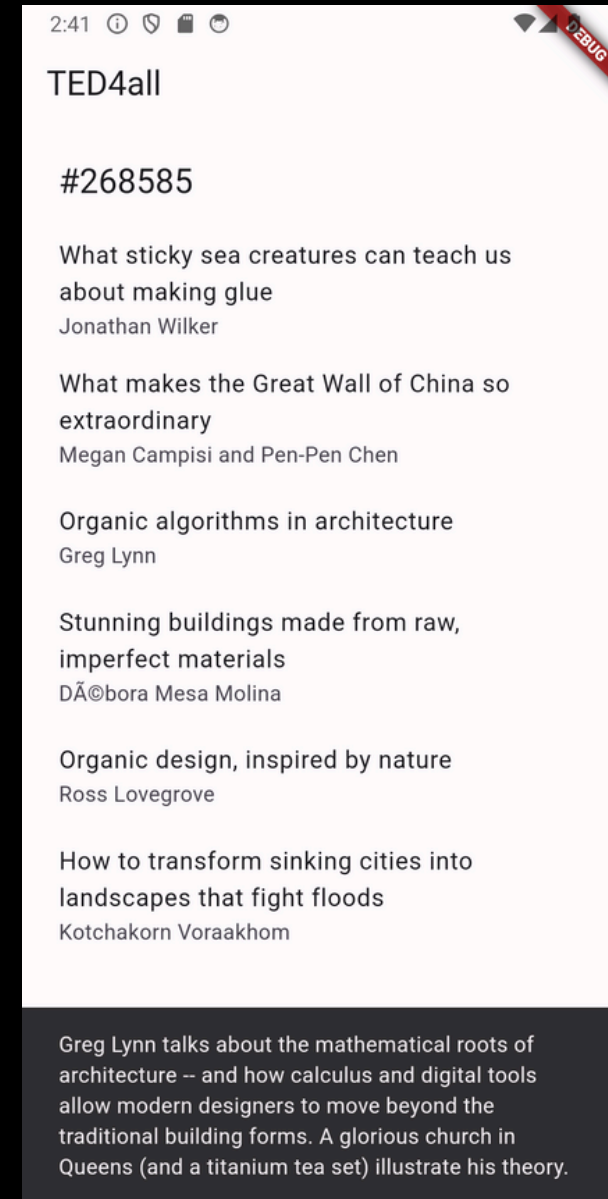
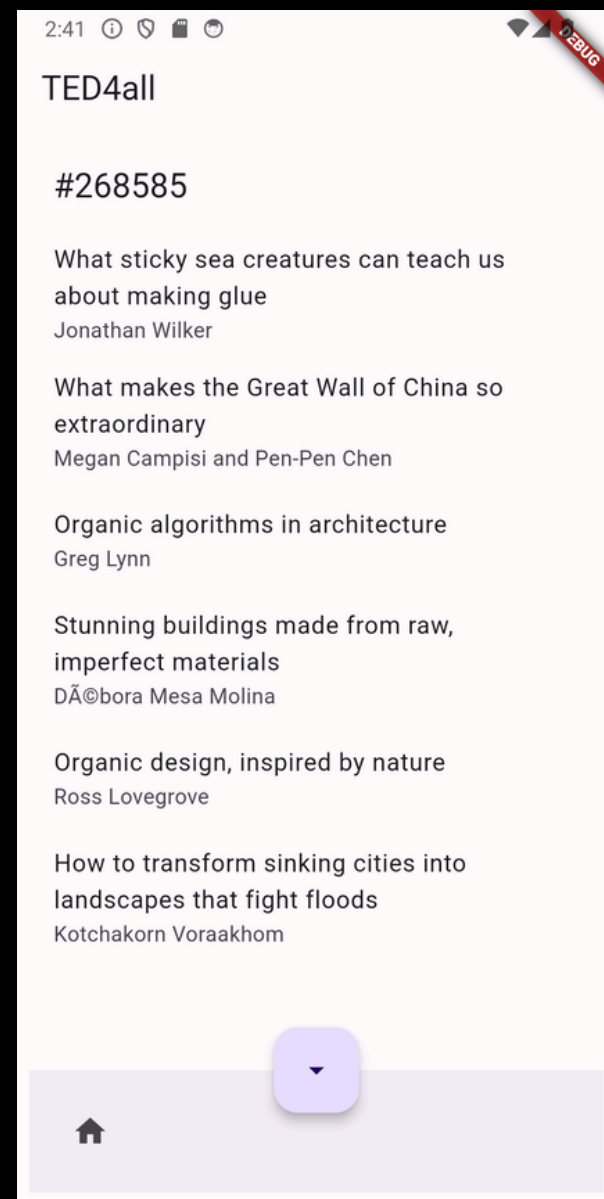


La schermata a sinistra rappresenta l'interfaccia di base sviluppata in classe, utilizzata principalmente per il testing delle funzionalità implementate. L'obiettivo è quello di realizzare un template simile alla schermata mostrata a destra nel mockup.

In questo mockup, è possibile osservare che i video consigliati verranno visualizzati in base ai video collegati all'ultimo talk guardato. Inoltre, tali video potranno essere filtrati per tag specifici. Nel caso in cui non ci siano video visualizzati né tag selezionati, verranno mostrati video casuali.



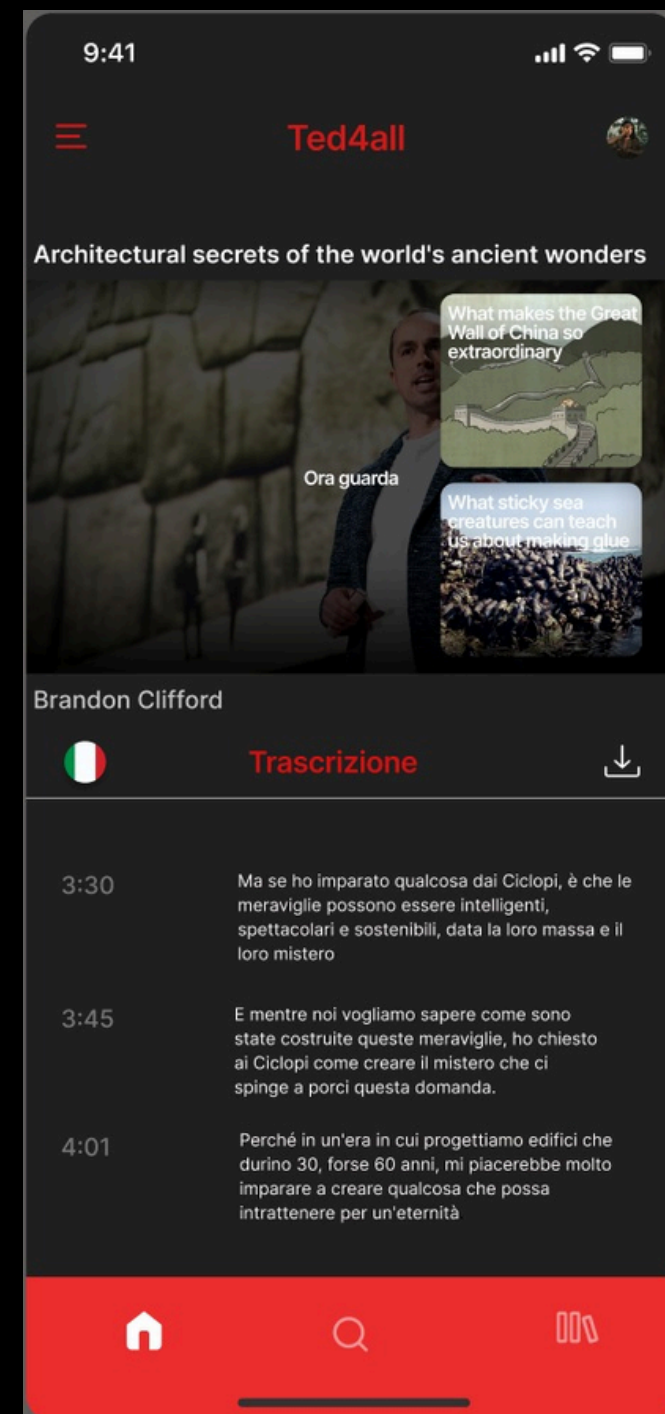
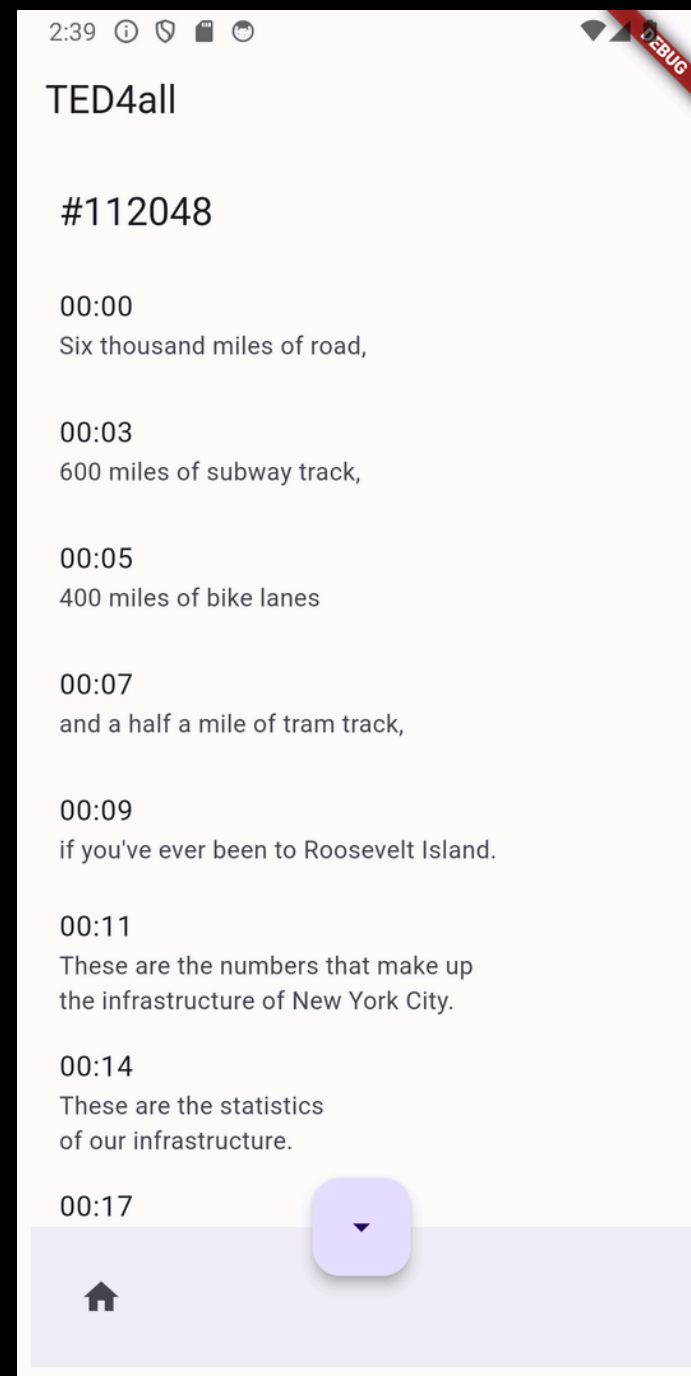
WATCH NEXT



```
class Talk {  
    final String title;  
    final String details;  
    final String mainSpeaker;  
    final String url;  
  
    Talk.fromJSON(Map<String, dynamic> jsonMap) :  
        title = jsonMap['title'],  
        details = jsonMap['description'],  
        mainSpeaker = (jsonMap['speaker'] ?? ""),  
        url = (jsonMap['url'] ?? "");  
}  
  
class RelatedVideos {  
    final List<Talk> relatedVideos;  
  
    RelatedVideos.fromJSON(Map<String, dynamic> json) :  
        relatedVideos = (json['related_videos'] as List).map((i) => Talk.fromJSON(i)).toList();  
}
```

La funzione **Watch Next** verrà utilizzata come detto precedentemente nella schermata principale per i video consigliati. Inoltre come verrà mostrato nelle prossime slides verrà utilizzato anche alla fine di ogni video per suggerire due video correlati.

TRANSCRIPT



```
class Transcript {  
    final String time;  
    final String text;  
  
    Transcript.fromJSON(Map<String, dynamic> jsonMap) :  
        time = (jsonMap['time'] ?? ""),  
        text = (jsonMap['text'] ?? "");  
}  
  
class TranscriptList {  
    final List<Transcript> transcript;  
  
    TranscriptList.fromJSON(Map<String, dynamic> jsonMap) :  
        transcript = (jsonMap['transcript'] as List).map((i) => Transcript.fromJSON(i)).toList();  
}
```

La funzione **Transcript** verrà utilizzata come indicato nel markup ovviamente per la trascrizione e per i sottotitoli. Si potrà selezionare la lingua ed eventualmente scaricare il testo integrale.

La schermata mostrata indica la fine di un video nella quale vengono suggeriti i due video, come detto precedentemente.

CRITICITÀ E SVILUPPI FUTURI



Il processo di installazione di Flutter e dell'emulatore ha presentato diverse problematiche, complicando l'inizio del lavoro di sviluppo. Tuttavia, una volta superati questi ostacoli iniziali, le attività di debug si sono rivelate semplici ed efficaci. Questo ha contribuito a migliorare l'efficienza complessiva del processo di sviluppo, nonostante le difficoltà incontrate nella fase di configurazione iniziale.



Il linguaggio di programmazione Dart, essendo significativamente diverso da quelli utilizzati in passato, ha sollevato diverse problematiche di apprendimento e adattamento. Lo sviluppo del layout, sebbene sarebbe stato facilitato dal sistema di widget di Flutter, avrebbe richiesto un notevole dispendio di tempo e risorse.



TED4all

Ponte tra culture e lingue



[Trello Board](#)



[GitHub Repository](#)

