

Project 4 – Bhagavad Gita App

Anirudh Belwadi (Andrew ID: abelwadi)

Description:

My application provides users access to chapters and verses from the Bhagavad Gita, along with their meanings and translations. The Android app makes requests to a custom Java servlet-based web service, which in turn fetches data from a 3rd-party Bhagavad Gita API. The service also logs usage to a MongoDB database, which is used to generate analytics on a web dashboard.

1. Implement a Native Android Application

a. Has at least three different kinds of views in the Layout

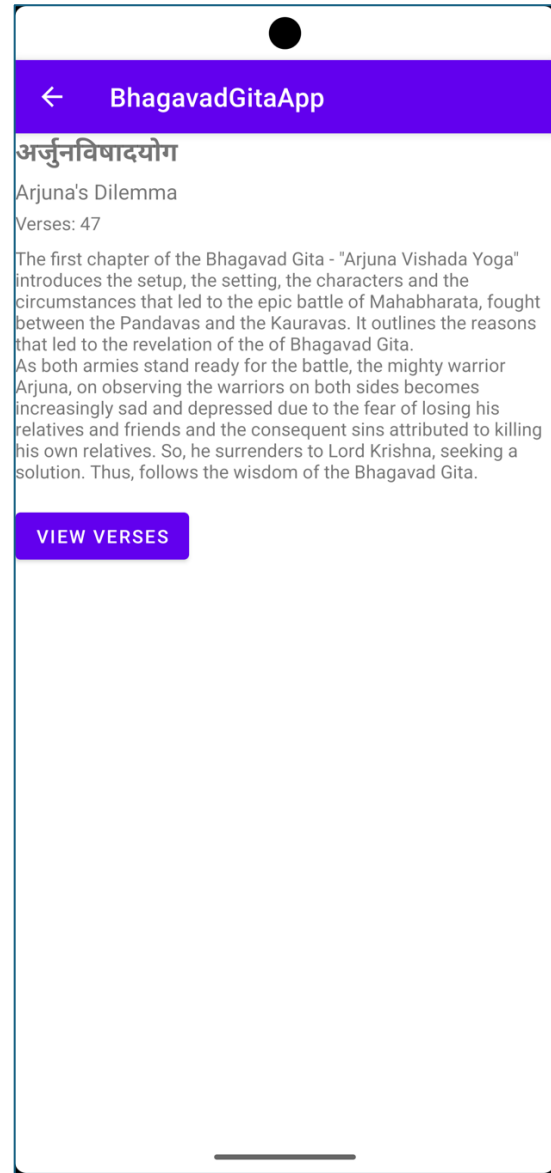
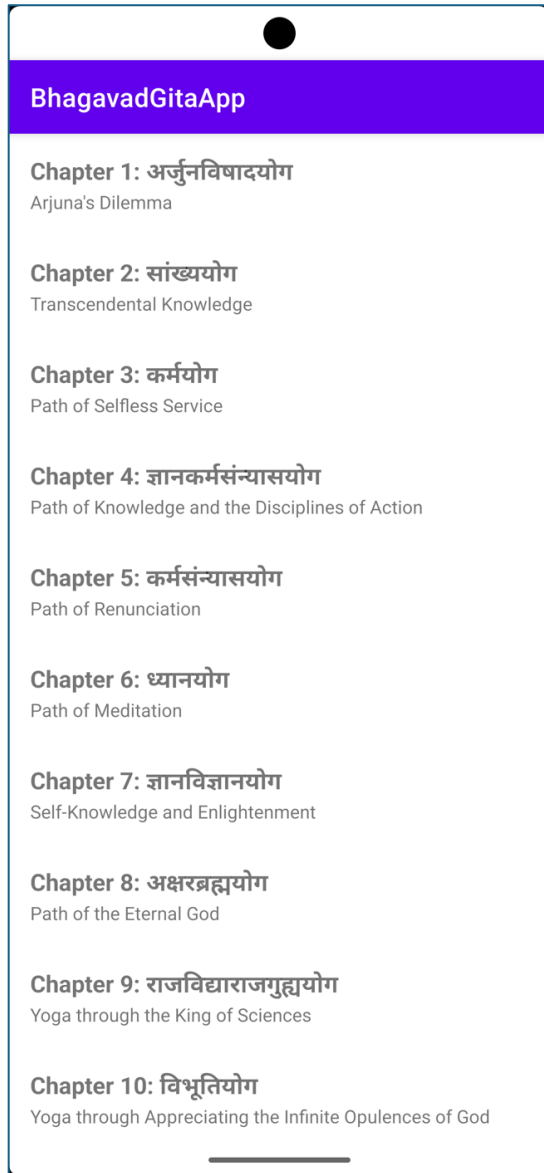
TextView (for names, summaries)

Button (View Verses)

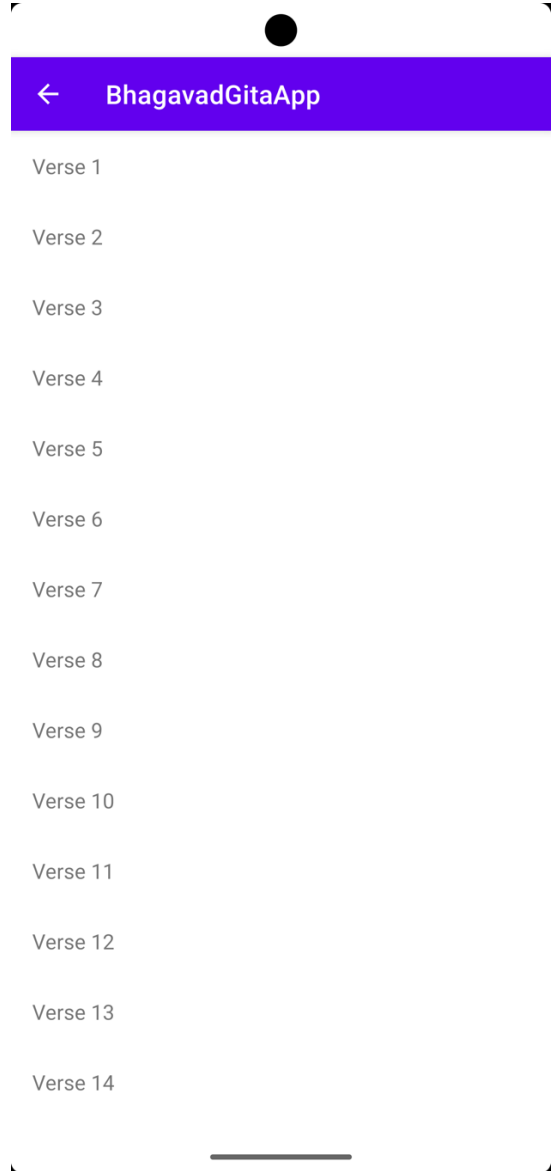
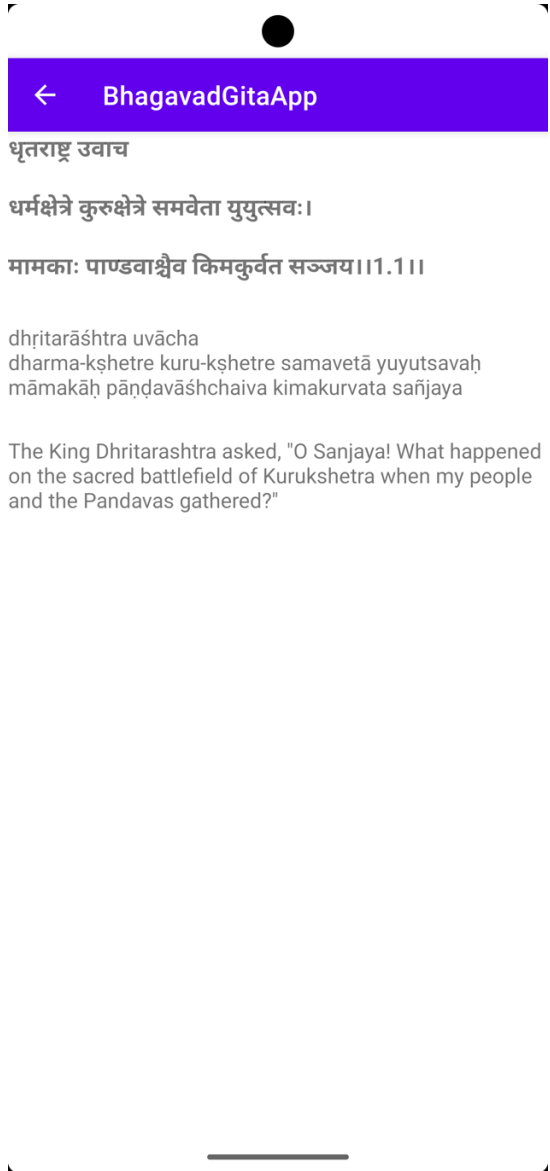
RecyclerView (Lists)

ScrollView (for detail pages)

Chapter List And Chapter Description Screens



Verse List And Verse Description Screens

	
Verse 1	धृतराष्ट्र उवाच धर्मक्षेत्रे कुरुक्षेत्रे समवेता युयुत्सवः। मामकाः पाण्डवाश्चैव किमकुर्वत सञ्जय॥1.1॥ dhṛitarāśhtra uvācha dharma-kṣhetre kuru-kṣhetre samavetā yuyutsavaḥ māmakāḥ pāṇḍavāśhchaiva kimakurvata sañjaya The King Dhritarashtra asked, "O Sanjaya! What happened on the sacred battlefield of Kurukshetra when my people and the Pandavas gathered?"
Verse 2	
Verse 3	
Verse 4	
Verse 5	
Verse 6	
Verse 7	
Verse 8	
Verse 9	
Verse 10	
Verse 11	
Verse 12	
Verse 13	
Verse 14	

b. Requires input from the user

The user interacts by selecting chapters and verses.

c. Makes an HTTP request (using an appropriate HTTP method) to your web service

The Android app uses Retrofit 2 as the HTTP client and defines four endpoints in the ApiService.java interface. These endpoints correspond to different levels of detail users can access in the Bhagavad Gita.

Each call is made through the ApiClient class, which configures the Retrofit base URL pointing to the web service: <https://friendly-fiesta-vgxx45x49542w54j-8080.app.github.dev>

Endpoint 1: Get All Chapters**URL Pattern:**

/api/chapters

Trigger in App:

When ChapterListActivity starts

Android Call:

```
api.getChapters().enqueue(new Callback<JsonObject>() { ... });
```

Backend Flow:

Android sends request to backend

Backend calls:

<https://bhagavad-gita3.p.rapidapi.com/v2/chapters/?skip=0&limit=18>

Backend returns a simplified list of chapters as JSON to the app

Endpoint 2: Get Chapter Details

URL Pattern:

/api/chapters/{chapterId}

Trigger in App:

When a user clicks on a chapter item

Android Call:

```
api.getChapterDetails(chapterId).enqueue(new Callback<JsonObject>() { ... });
```

Backend Flow:

App sends chapterId (e.g., 1)

Backend calls:

<https://bhagavad-gita3.p.rapidapi.com/v2/chapters/1>

Backend filters response and returns only name, meaning, summary, and verse count to the app

Endpoint 3: Get All Verses in Chapter

URL Pattern:

/api/chapters/{chapterId}/verses

Trigger in App:

When user taps “View Verses” in chapter detail screen

Android Call:

```
api.getVerses(chapterId).enqueue(new Callback<JsonObject>() { ... });
```

Backend Flow:

App sends chapterId to the backend

Backend calls:

<https://bhagavad-gita3.p.rapidapi.com/v2/chapters/1/verses/>

Backend returns simplified verse list: verse numbers and slugs

Endpoint 4: Get Verse Details

URL Pattern:

/api/chapters/{chapterId}/verses/{verseld}

Trigger in App:

When a user taps on a specific verse

Android Call:

```
api.getVerseDetails(chapterId, verseNumber).enqueue(new Callback<JsonObject>() { ... });
```

Backend Flow:

App sends both chapterId and verseNumber

Backend calls:

<https://bhagavad-gita3.p.rapidapi.com/v2/chapters/1/verses/1>

Backend filters and returns:

Original Sanskrit verse

Transliteration

English translations (from various authors)

Common Data in Requests:

All requests include custom headers:

X-Device-Model

X-Device-OS

All logs are saved to MongoDB for analytics

d. Receives and parses a JSON reply from the web service

All Android logic uses Retrofit and Gson to parse custom DTOs.

e. Displays new information to the user

Users see summaries, verse details, and translations dynamically.

f. Is repeatable

The user can navigate across screens repeatedly without app restarts. Back stack is preserved.

2. Implement a Web Application (Backend Service)

The URL of my web service deployed to codespace is:

<https://friendly-fiesta-vgxx45x49542w54j-8080.app.github.dev>

The GitHub repo name is [distributed-systems-project-04-anirudhbelwadi](#)

a. Using an HttpServlet to implement a simple (can be a single path) API

In my web app project:

Models are placed in the ***project4backendservice.backendservice.model*** package

Views are placed in the ***webapp*** folder

Controllers are placed in ***project4backendservice.backendservice.controller*** package

b. Receives an HTTP request from the native Android application

The Android app sends GET requests to the servlet endpoints using Retrofit. Each request includes:

URL path (e.g., /api/chapters/1)

HTTP headers:

X-Device-Model

X-Device-OS

On the server, I extract those using:

```
String model = request.getHeader("X-Device-Model");
```

```
String os = request.getHeader("X-Device-OS");
```

And I extract query/path parameters from the URL.

c. Executes business logic appropriate to your application

The servlet parses the request, then:

1. Builds a URL to the Bhagavad Gita API on RapidAPI
2. Makes an HTTP request using Java HttpClient
3. Parses the response JSON
4. Filters out only required fields
5. Logs details to MongoDB
6. Sends a clean JSON response back to the Android app

d. Replies to the Android application with a JSON formatted response.

This is a sample of a JSON sent by the endpoint GET /api/chapters/1. The servlet forwards a request to the RapidAPI - Bhagavad Gita API, parses the response, and extracts only the most relevant fields to return to the Android client.

```
{  
  "name": "अर्जुनविषादयोग",  
  "meaning": "Arjuna's Dilemma",  
  "summary": "The first chapter of the Bhagavad Gita - \"Arjuna  
Vishada Yoga\" introduces the setup, the setting, the characters and  
the circumstances that led to the epic battle of Mahabharata...",  
  "verses_count": 47  
}
```


4. Log Useful Information

I have logged the following fields in my MongoDB logs collection. Each was chosen with a clear purpose to support future analytics, monitoring, and debugging for the mobile application:

- Event:
 - This string describes what type of action the user performed, such as `get_chapters`, `get_verse_details`, etc. It helps group and filter different API usage patterns, which is useful for:
 - Understanding which app features are most used
 - Measuring endpoint popularity
 - Tracking potential misuse or unexpected traffic patterns
- Timestamp
 - The exact time when the request was received by the backend, stored as a Unix epoch. This allows:
 - Creating visualizations like request volume over time
 - Auditing when a request happened
 - Correlating logs with performance bottlenecks
- Source
 - Currently hardcoded as `"android_app"` but designed to be extensible in case the app is later ported to iOS or used via web. This allows:
 - Differentiating between platforms
 - Understanding OS-specific traffic patterns
 - Filtering logs by client type in analytics dashboards
- Device Model
 - Captured from the Android app using `Build.MODEL`. This helps:
 - Identify which devices are most commonly used
 - Detect device-specific bugs (e.g., rendering issues or API failures)
 - Plan optimizations for popular device classes (e.g., budget phones vs. flagships)
- OS Version
 - Captured using `Build.VERSION.RELEASE`. This is useful for:
 - Understanding adoption of different Android versions
 - Ensuring compatibility across OS releases
 - Prioritizing testing efforts for dominant OS versions
- Request params (chapter/verse)
 - An array of key-value strings like `"chapter_id=4"` or `"verse_id=4"`. This shows exactly what data the client requested, enabling:

- Easy reproduction of bugs (you know what was queried)
- Filtering logs based on chapter or verse usage
- Tracking which Gita chapters/verses are most accessed by users

Sample MongoDB document:

```
{
  "_id": {
    "$oid": "67f7fa1d4152e46e868e50b9"
  },
  "event": "get_chapters",
  "timestamp": {
    "$numberLong": "1744304669497"
  },
  "source": "android_app",
  "model": "sdk_gphone64_arm64",
  "os": "Android 16",
  "request_params": [
    "chapter_id=4",
    "verse_id=4"
  ]
}
```

5. Store the log information in a database

Connection String:

mongodb+srv://abelwadi:eg87g3edb122@mydbcluster.xqjeg28.mongodb.net/?retryWrites=true&w=majority&appName=MyDBCluster

6. Display operations analytics and full logs on a web-based dashboard

API Usage Dashboard

Event	Timestamp	Device Model	OS Version	Params
get_chapters	Thu Apr 10 16:10:58 UTC 2025	null	null	
get_chapters	Thu Apr 10 16:13:36 UTC 2025	null	null	
get_chapter_details	Thu Apr 10 16:13:44 UTC 2025	null	null	chapter_id=1
get_chapters	Thu Apr 10 16:15:12 UTC 2025	null	null	
get_chapter_details	Thu Apr 10 16:15:15 UTC 2025	null	null	chapter_id=1
get_verse_list	Thu Apr 10 16:15:25 UTC 2025	null	null	chapter_id=1
get_verse_details	Thu Apr 10 16:15:38 UTC 2025	null	null	chapter_id=1, verse_id=3
get_chapters	Thu Apr 10 16:15:53 UTC 2025	null	null	
get_chapter_details	Thu Apr 10 16:16:03 UTC 2025	null	null	chapter_id=8
get_verse_list	Thu Apr 10 16:16:11 UTC 2025	null	null	chapter_id=8
get_verse_details	Thu Apr 10 16:16:17 UTC 2025	null	null	chapter_id=8, verse_id=13
get_chapters	Thu Apr 10 16:32:29 UTC 2025	null	null	
get_chapters	Thu Apr 10 16:32:48 UTC 2025	null	null	
get_chapter_details	Thu Apr 10 16:32:51 UTC 2025	null	null	chapter_id=1
get_chapters	Thu Apr 10 16:33:33 UTC 2025	null	null	
get_chapter_details	Thu Apr 10 16:33:36 UTC 2025	null	null	chapter_id=1
get_chapter_details	Thu Apr 10 16:33:40 UTC 2025	null	null	chapter_id=4
get_verse_list	Thu Apr 10 16:33:47 UTC 2025	null	null	chapter_id=4
get_verse_details	Thu Apr 10 16:33:50 UTC 2025	null	null	chapter_id=4, verse_id=4
get_chapters	Thu Apr 10 17:04:29 UTC 2025	sdk_gphone64_arm64	Android 16	
get_chapters	Thu Apr 10 17:54:02 UTC 2025	sdk_gphone64_arm64	Android 16	
get_chapters	Thu Apr 10 19:13:56 UTC 2025	sdk_gphone64_arm64	Android 16	
get_chapter_details	Thu Apr 10 19:14:17 UTC 2025	sdk_gphone64_arm64	Android 16	chapter_id=1
get_verse_list	Thu Apr 10 19:14:29 UTC 2025	sdk_gphone64_arm64	Android 16	chapter_id=1
get_verse_details	Thu Apr 10 19:14:45 UTC 2025	sdk_gphone64_arm64	Android 16	chapter_id=1, verse_id=1