

**Sri Lanka Institute of Information  
Technology**

**Programming Applications and Frameworks (IT3030)**

**Final Assignment Report**

**GROUP ID: DS-20**



**CosmoConnect – A Space & Astronomy Skill-Sharing and Learning Web Application**

<b>Student ID</b>	<b>Student Name</b>
IT22178640	Kumar Dayana Priyadarshani
IT22057488	Elangovan Harishalinee
IT22294098	Prabagaran Vidursha
IT22289384	Nagenthraraja Kaushalya

Date: 07.05.2025

## Table of Contents

1.	Introduction .....	3
2.	Functional Requirements & Non-Functional Requirements.....	4
	Functional Requirements .....	4
	Non-Functional Requirements .....	7
3.	Overall Architecture Diagram.....	8
4.	REST API Architecture diagram.....	8
5.	Frontend Architecture Diagram .....	9
6.	System Functions .....	10
	Post, Comment and Like CRUD Functionality – IT22178640.....	10
	Learning Progress CRUD Functionality - IT22057488.....	23
	Competition Management CRUD Functionality - IT22294098 .....	32
	Learning Plan CRUD Functionality - IT22289384 .....	40
	User Profile Management CRUD functionality – Group Work .....	48
7.	GitHub.....	53
8.	References .....	59

# 1. Introduction

Cosmo Connect is a community skill sharing web application for astronomers to study, share ideas, and engage through organized and interactive learning. The application invites the users to submit their night sky experience, astrophotography, home-space projects, and offer personalised study materials and astronomy competitions. Under this system, students can engage with each other, track progress, and practice skill-builder exercises. The primary goal is to make access to astronomy and its appeal more attractive for students, amateur astronomers, and space science enthusiasts globally.

This system was developed because of the lack of skill-sharing, practical web applications for astronomy. While many theoretical resources are available, few websites allow for hands-on learning through interaction with a community. Cosmo Connect bridges this gap by allowing users to actively contribute to astronomy-based content creation and learning in a social environment. The platform is developed for students, amateur astronomers, hobbyists, teachers, and interested users interested in learning space science. The app offers an environment for users to learn, contribute, and work together regardless of how much astronomy background they have. By integrating the use of peer interaction, ordered learning, and competitions, the app invites users to continue improving and donate their knowledge accordingly.

This application was implemented using a modern tech stack for performance and scalability. The frontend was developed using ReactJS (via Visual Studio Code), and the backend was set up using Spring Boot (via IntelliJ IDEA Community Edition). MongoDB is used as the database, Node.js assists runtime functionality, and Maven is used for dependency management. Postman was employed for API testing, and the development was sustained using GitHub Desktop for version control and collaboration.

These are the fundamental functions relating to our web application such as Post and Comment Management, Learning Progress Tracker, Learning Plan Management, Competition Participation and Management.

This platform has been carefully designed keeping in mind functional needs and user experience, supported by robust architecture and modern development techniques. Design process, system components, and implementation logic are also elaborated further within the document and offer more information about how the platform was designed, built, and maintained.

## 2. Functional Requirements & Non-Functional Requirements

### Functional Requirements

#### **Space Skill Showroom (Skill Sharing Posts)**

- Allow users to showcase space-related skills by uploading photos or short videos (up to 30 seconds).
- Allow users to share content such as astrophotography, DIY telescopes, space science experiments, and creative builds.
- Allow users to add detailed descriptions, titles, and select appropriate categories for each post.
- Allow users to tag experienced contributors as mentors to provide guidance through comments or messages.
- Allow AI to analyze uploaded content, recommend learning resources, and suggest improvements.
- Allow the system to automatically generate relevant space-related tags based on post content.
- Allow users to like, comment on, and share posts within the platform.
- Allow users to collaborate on a single post with shared editing permissions.
- Allow users to create interactive polls and mini-challenges to engage the community.
- Allow users to group posts into playlists or series to present learning or project progressions.
- Allow users to search and filter posts by tags, categories, contributors, and engagement trends.
- Allow users to upload media to cloud storage with automatic compression and validation.
- Allow users and admins to view analytics related to post performance and overall engagement.

#### **Galactic Learning Roadmap (Learning Plan Sharing)**

- Allow users to upload and share structured learning plans that include phases, topics, objectives, timelines, and recommended resources.
- Allow users to create and customize learning plans with editable titles, topics, timelines, goals, and resource links.
- Allow users to attach curated resources such as tutorial videos, PDFs, toolkits, and reference links to each phase or topic within the learning plan.
- Allow users to add a visual mission-style layout to enhance the engagement and presentation of their learning plans.

- Allow users to embed preview summaries of learning plans into posts, comments, or profiles to increase visibility and inspire exploration.
- Allow users to earn digital badges or recognition when their learning plans reach milestones such as views, shares, or completions.
- Allow users to attach quizzes and assessments to each learning plan and issue certificates upon successful completion.
- Allow users to invite peers or mentors to co-learn, edit, and contribute to their learning plans.
- Allow users to edit and delete learning plans, including individual topics, attached resources, or deadline entries.
- Allow users to receive AI-powered personalized weekly learning boosters that suggest micro-tasks, motivational quotes, and quick challenges based on their current learning pace and topics.

## **Space Learning Journey (Learning Progress Updates)**

- Allow users to log and track their learning progress with titles, descriptions, categories, and optional media.
- Allow users to manage the visibility of each log entry, keeping them private by default or marking milestones as public.
- Allow users to edit, delete, and update their learning log entries to reflect current progress or remove outdated information.
- Allow users to earn and display skill badges based on learning achievements, consistency, and milestone completions.
- Allow AI to analyze personal learning logs and recommend personalized learning paths, topics, or resources.
- Allow users to access a dedicated “Learning Journey” tab within their profile that separates private logs from public milestones and badges.
- Allow users to share selected milestone entries with space-learning groups or communities based on relevant topics.
- Allow the backend to track logs, badge achievements, visibility status, group links, and AI recommendation data efficiently.

## **Personalized User Profiles**

- Allow users to create and personalize profiles with names, avatars, cosmic themes, and skill interests.
- Allow users to progress through a Cosmic Title System that upgrades their rank based on engagement.
- Allow users to display their contributions through skill-sharing posts, learning milestones, and contest results.
- Allow users to follow others and view a tailored space feed with updates from their network.
- Allow users to control privacy settings for learning logs, contest activity, and follower visibility.
- Allow users to activate a Spotlight Profile Mode to highlight expertise in selected skill areas.
- Allow users to securely log in using OAuth 2.0 with Google or Facebook with optional 2FA.
- Allow users to receive a Space Passport that logs key moments like joining, first post, and first contest.
- Allow users to update profile information, edit personal details, and delete accounts with secure data handling.

## **Competition Management (Admin Panel)**

- Allow admins to create and configure competitions with relevant details such as title, category, description, and competition format.
- Allow admins to define or update competition timelines including the start date, submission deadline, and result announcement.
- Allow admins to upload supporting materials such as banners, documents, datasets, and additional learning resources.
- Allow admins to assign judges, revise judging criteria, and make adjustments to rules or FAQs as needed.
- Allow admins to arrange and schedule Q&A sessions to support participant engagement and clarification.
- Allow admins to cancel or archive competitions by marking them appropriately without permanent removal.
- Allow admins to enable filtered browsing of competitions based on category, level, status, or featured tags.
- Allow admins to activate countdown displays for deadlines and maintain leaderboards to reflect participant rankings.

## Non-Functional Requirements

### 1. Performance & Scalability

Platform must be capable of handling a large number of concurrent users efficiently. It should provide a seamless experience, even when multiple users are uploading, streaming, or interacting simultaneously. The system should be designed with load balancing and caching mechanisms to ensure high performance.

### 2. Security & Authentication

User data security is a priority, requiring encryption for stored information and secure authentication methods. The platform should implement OAuth 2.0 for login and offer two factor authentication (2FA) to protect user accounts from unauthorized access. Compliance with security standards such as SSL/TLS for data transmission is essential.

### 3. Usability & User Experience (UX)

The platform should be intuitive, visually appealing, and easy to navigate. It must provide a responsive design that adapts to various devices, ensuring a smooth user experience on desktops, tablets, and mobile phones. Accessibility features should also be considered for users with disabilities.

### 4. Reliability & Availability

The system should be available 24/7 with minimal downtime. Redundant server setups, cloud-based hosting, and automated failover mechanisms should be implemented to ensure high availability and resilience in case of failures or unexpected surges in traffic.

### 5. Maintainability & Modularity

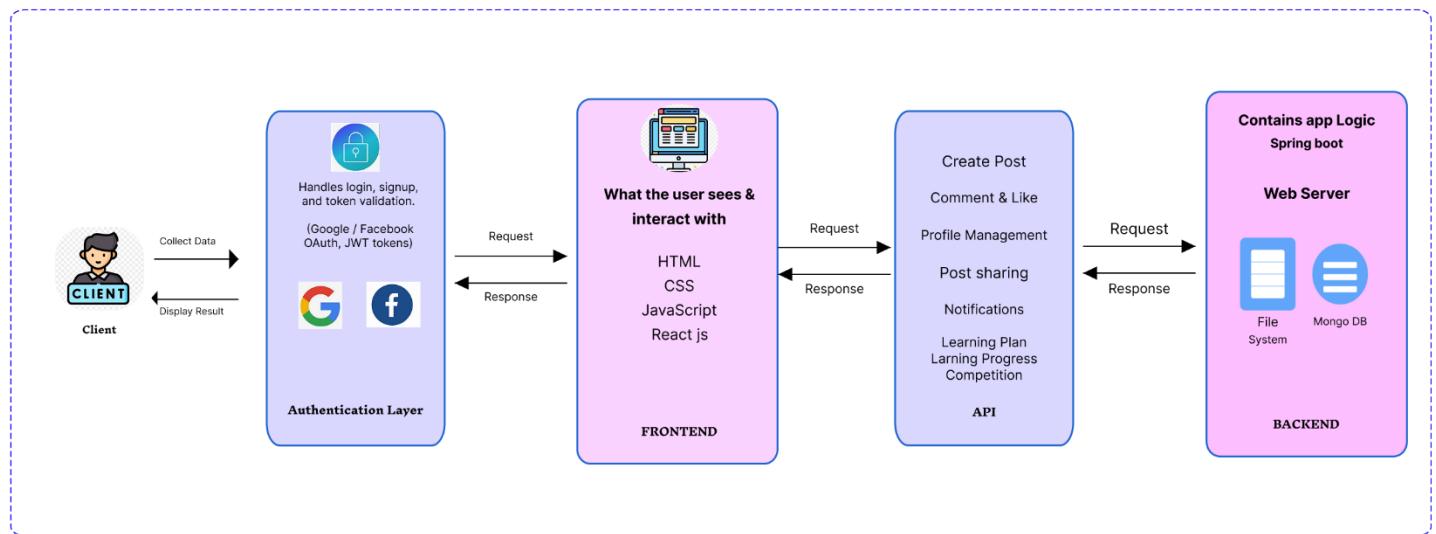
The software should be developed with a modular architecture, allowing for easy updates and enhancements without disrupting core functionalities. This ensures that new features, bug fixes, and security patches can be deployed efficiently without affecting user experience.

### 6. Data Integrity & Consistency

All user data, including progress tracking, learning plans, and interactions, must be stored reliably. Transactions should be atomic, ensuring that incomplete or inconsistent data does not occur due to system failures. Backup mechanisms should be in place to restore data in case of unexpected loss.

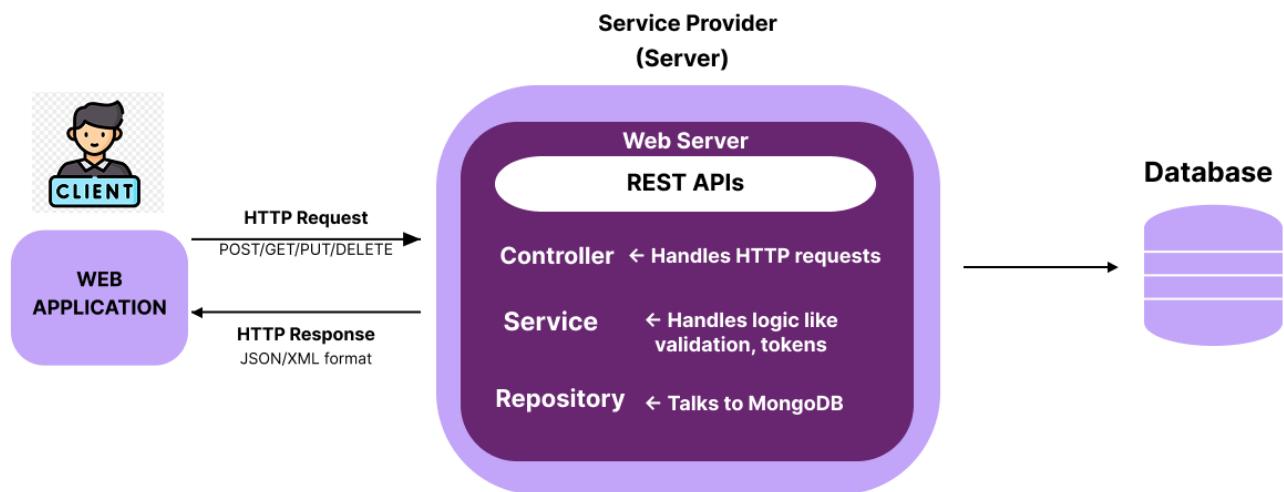
### 3. Overall Architecture Diagram

Overall Architecture Diagram for Entire System



### 4. REST API Architecture diagram

Architecture Diagram for REST API



## 5. Frontend Architecture Diagram

### Frontend Architecture Diagram



## 6. System Functions

### Post, Comment and Like CRUD Functionality – IT22178640

#### Function Overview

The Post and Comment module empowers users to showcase and share their space-related skills, experiments, and discoveries within a collaborative and engaging platform. Users can create visually rich posts by uploading images or short videos (up to 30 seconds), adding titles, detailed descriptions, selecting categories, and tagging mentors. This module supports interactive features such as likes, comments, shares, collaborative edits, and the creation of polls or mini-challenges. Comments serve as a medium for mentorship, feedback, and discussion. The system enhances discoverability through auto-generated tags and advanced search filters. Overall, this module boosts community learning and creative expression in space science.

#### User Roles & Permissions

##### ❖ Registered Users

- Can create, edit, and delete their own posts and comments.
- Can upload media (photos/videos), write descriptions, and categorize posts.
- Can tag mentors, like, comment on, and share others' posts.
- Can participate in polls, mini-challenges, and collaborative posts with shared editing rights.
- Can search and filter posts based on categories, tags, and contributors.

##### ❖ Mentors (Tagged by Users)

- Can provide guidance via comments on posts where they're tagged.
- Cannot directly modify posts unless granted shared editing permission.

##### ❖ Admins

- Can manage inappropriate content, featured tags, and engagement analytics.
- Can approve or remove collaborative editing permissions between users.
- Can monitor poll/challenge participation and flag misuse of features.

#### Functionalities Covered (CRUD Operations)

##### POST

###### Create – What can users create?

- Upload images or videos (up to 30 seconds).
- Add title, description, and select relevant categories.
- Tag mentors for feedback and support.
- Enable collaborative editing by inviting contributors.
- Initiate polls or mini-challenges.
- Group related posts into playlists or learning series.

The screenshot shows a Postman request for a POST method to the URL `http://localhost:8080/api/files/upload`. The 'Body' tab is selected, showing a 'form-data' structure with a single field named 'File Upload' containing the file 'env.png'. Other options like 'x-www-form-urlencoded', 'raw', 'binary', and 'GraphQL' are available but not selected.

The screenshot shows a Postman request for a POST method to the URL `http://localhost:8080/api/posts`. The 'Body' tab is selected, showing a JSON payload for a new post. The JSON is formatted with line numbers and includes fields such as title, description, content, media URLs, media type, telescope used, location, celestial object, observation date/time, observation conditions, is public status, coordinates, exposure time, equipment details, and processing details. The 'JSON' dropdown at the bottom right is set to 'Beautify'.

```

1 {
2   "title": "My First Astronomical Observation",
3   "description": "Observing the Andromeda Galaxy",
4   "content": "Detailed observation notes...",
5   "mediaUrls": ["http://localhost:8080/api/files/filename.jpg"], // Use filename from upload response
6   "mediaType": "IMAGE",
7   "telescopeUsed": "Celestron NexStar 8SE",
8   "location": "Backyard Observatory",
9   "celestialObject": "Andromeda Galaxy",
10  "observationDateTime": "2024-03-20T20:00:00",
11  "observationConditions": "Clear skies, low light pollution",
12  "isPublic": true,
13  "coordinates": "00h 42m 44.3s +41° 16' 9\"",
14  "exposureTime": "30 seconds",
15  "equipmentDetails": "Celestron NexStar 8SE, Canon EOS 6D",
16  "processingDetails": "Stacked 10 images using DeepSkyStacker"
17 }

```

## Read – What data is viewable and by whom?

- All registered users can view published posts.
- Visible post elements include media, description, author, tags, category, and engagement metrics (likes/comments/shares).
- Posts are searchable by category, tags, author, or popularity trends.

The screenshot shows a Postman request for a GET method to the URL `http://localhost:8080/api/posts/{postId}`. The 'Body' tab is selected, showing a 'raw' JSON payload identical to the one in the previous screenshot. The 'JSON' dropdown at the bottom right is set to 'Beautify'.

## Update – What can be modified and under what conditions?

- Authors can update their own posts: media, title, description, tags, categories.
- Collaborators (if added) can edit shared posts.
- Admins can update any post for moderation or content correction.

The screenshot shows the Postman application interface. At the top, it says "HTTP http://localhost:8080/api/posts/{postId}". Below that, a dropdown menu shows "PATCH" selected. The URL field contains "http://localhost:8080/api/posts/{postId}". To the right of the URL are "Save" and "Share" buttons. Underneath the URL, there are tabs: "Params", "Authorization", "Headers (9)", "Body", "Scripts", "Tests", and "Settings". The "Body" tab is currently active, indicated by a red underline. Below the tabs, there are options: "none", "form-data", "x-www-form-urlencoded", "raw", "binary", "GraphQL", and "JSON". "JSON" is selected, indicated by a blue outline. The "Body" section contains the following JSON code:

```
1
2
3   "title": "Updated Title",
4   "description": "Updated description"
5
```

## Delete – What can be removed, and are there any restrictions?

- Only the post creator can delete their post.
- Admins can delete posts that violate platform policies.
- Deletion may be soft (hidden) for moderation records.

The screenshot shows the Postman application interface. At the top, it says "HTTP http://localhost:8080/api/posts/{postId}". Below that, a dropdown menu shows "DELETE" selected. The URL field contains "http://localhost:8080/api/posts/{postId}". To the right of the URL are "Save" and "Share" buttons. Underneath the URL, there are tabs: "Params", "Authorization", "Headers (9)", "Body", "Scripts", "Tests", and "Settings". The "Body" tab is currently active, indicated by a red underline. Below the tabs, there are options: "none", "form-data", "x-www-form-urlencoded", "raw", "binary", "GraphQL", and "JSON". "JSON" is selected. The "Body" section is empty.

## COMMENT

### Create – What can users create?

- Add comments on any public post.
- Mentors can leave guidance/comments on tagged posts.
- Users can reply to existing comments if thread-based replies are enabled.

The screenshot shows the Postman application interface. At the top, it says "HTTP http://localhost:8080/api/comments". Below that, a dropdown menu shows "POST" selected. The URL field contains "http://localhost:8080/api/comments". To the right of the URL are "Save" and "Share" buttons. Underneath the URL, there are tabs: "Params", "Authorization", "Headers (9)", "Body", "Scripts", "Tests", and "Settings". The "Body" tab is currently active, indicated by a red underline. Below the tabs, there are options: "none", "form-data", "x-www-form-urlencoded", "raw", "binary", "GraphQL", and "JSON". "JSON" is selected. The "Body" section contains the following JSON code:

```
1
2   "content": "Great observation!",
3   "postId": "{postId}"
4
```

## Read – What data is viewable and by whom?

- All registered users can view comments under each post.
- Comments display commenter's name, timestamp, and message.
- Nested replies (if enabled) are visible to all viewers.

The screenshot shows a REST client interface. At the top, it displays the URL `http://localhost:8080/api/comments/post/{postId}`. Below the URL, there is a method dropdown set to `GET` and a "Send" button. The "Headers" tab is selected, showing 9 items. The "Body" tab is also selected. At the bottom, there are options for "none", "form-data", "x-www-form-urlencoded", "raw", "binary", "GraphQL", and "JSON".

## Update – What can be modified and under what conditions?

- Users can edit their own comments within a short window (e.g., 10 minutes).
- Admins can edit any comment for moderation reasons.

The screenshot shows a REST client interface. At the top, it displays the URL `http://localhost:8080/api/comments/post/{postId}`. Below the URL, there is a method dropdown set to `PATCH` and a "Send" button. The "Headers" tab is selected, showing 9 items. The "Body" tab is also selected. At the bottom, there are options for "none", "form-data", "x-www-form-urlencoded", "raw", "binary", "GraphQL", and "JSON".

## Delete – What can be removed, and are there any restrictions?

- Users can delete their own comments.
- Admins can delete any comment flagged as inappropriate.
- Deleted comments are hidden or soft-deleted for record-keeping.

The screenshot shows a REST client interface. At the top, it displays the URL `http://localhost:8080/api/comments/post/{postId}`. Below the URL, there is a method dropdown set to `DELETE` and a "Send" button. The "Headers" tab is selected, showing 9 items. The "Body" tab is also selected. At the bottom, there are options for "none", "form-data", "x-www-form-urlencoded", "raw", "binary", "GraphQL", and "JSON".

## Like & Unlike Functionality (Interaction Features – Not CRUD)

- Users can like any public post once; this action increases the like count.
- If a post is already liked, clicking again will unlike it (toggle action).
- The system keeps track of which users liked which posts (for analytics or engagement features).
- Like status (liked/unliked) is stored per user and reflected in the UI (e.g., filled vs. outlined heart icon).

## User Interface Design (UI Screenshots)

- Post Creation UI with Poll creation option and playlist option

The screenshot displays the COSMO CONNECT application interface. At the top, there's a navigation bar with icons for Home, Learn, Compete, Explore, and a profile icon. The main area shows a user profile on the left for "Astro Explorer". The profile includes a circular profile picture, the name "Astro Explorer", a bio about exploring the cosmos and learning about deep space objects and astrophotography, a learning progress bar at 75%, and follower counts of 128 Followers and 64 Following. Below the profile, under "Skills", are listed Telescope Operation, Astrophotography, and Star Mapping. To the right, there's a post creation section. It features a text input field with placeholder text about a telescope setup, a "Create a Poll" section with two options ("Option 1" and "Option 2") and a "Add Option" button, and an "Add to Learning Playlist" section with three categories: "Telescope" (highlighted with a cursor), "Star", and "Deep". Below these sections are buttons for "Posts", "Progress", and "Community". The bottom part of the screenshot shows a feed of two posts from "Astro Explorer". The first post, made 2h ago, says "Just captured the Orion Nebula! #astrophotography" with 45 likes and a share button. The second post, also made 2h ago, says "Learning about variable stars today. Fascinating stuff! ⭐" with 32 likes and a share button.

- Post Read UI (Displaying User's Own Post)

The screenshot shows the Cosmo Connect mobile application interface. At the top, there's a navigation bar with icons for Feed, Learn, Compete, Explore, and a user profile. Below the navigation is a dark-themed header with a user profile picture, the name "Astro Explorer", and a text input field that says "Share your cosmic discoveries...". There are also icons for a globe, moon, calendar, and other navigation options, along with "Collaborate" and "Launch Post" buttons.

The main content area has tabs for Posts, Progress, and Community. A post by "Astro Explorer" is displayed, featuring a large image of a telescope setup against a sunset sky. The post text reads:

This photo captures my latest telescope setup under a beautifully clear sky just after sunset. I used a Newtonian reflector on an equatorial mount to begin my deep sky journey. The alignment was perfect, and I managed to track several constellations including Orion and Cassiopeia. This setup marks a major step in my astrophotography learning journey, and I'm excited to explore more targets like the Andromeda Galaxy and star clusters in the coming nights.

Below the post are interaction icons for a lightbulb (1), a bookmark, and a comment, followed by a timestamp "2h ago".

- Delete Page UI for Post

The screenshot shows the same Cosmo Connect interface as above, but with a modal dialog box overlaid on the post. The dialog has a purple rocket icon and the text "Delete Post?". It asks, "Are you sure you want to delete this post? This action cannot be undone." There are "Cancel" and "Delete" buttons. The background post content is partially visible.

- All users posts Feed with story option

The screenshot displays the COSMO CONNECT mobile application's feed section. At the top, there is a navigation bar with icons for Home, Feed, Learn, Compete, Explore, Posts, and Profile.

**Left Column:**

- Mini Space Quiz:** A card asking "Which planet is known as the Red Planet?" with options: Venus, Mars, Jupiter, and Saturn. It includes a globe icon.
- Daily Space Challenge:** A circular button with the text "Draw or photograph your favorite constellation."
- Spin for New Challenge:** A button with a star icon.
- Space Calendar:** A list of celestial events with dates and descriptions:
  - Jul 21: Delta Aquariids Meteor Shower
  - Aug 12: Perseid Meteor Shower
  - Aug 19: Blue Moon
  - Sep 18: Neptune at Opposition
  - Oct 14: Annular Solar Eclipse
  - Nov 17: Leonid Meteor Shower
  - Dec 14: Geminid Meteor Shower

**Top Center:** A row of user profile icons: Add Story, Stellar, Luna, Comet, Rocket, and Aurora.

**Middle Center:** A post by "Stellar Sam" (1h ago) featuring a photo of the Milky Way core. The caption reads: "Captured the Milky Way core from a dark sky site. The clarity tonight was out of this world! 🌌". Below the photo are interaction icons: thumbs up, bookmark, comments (5), and shares (2).

**Right Column:**

- Suggested Users to Follow:** A list of profiles with "Follow" buttons:
  - Astro Alex (Astrophotographer)
  - Luna Lee (Lunar Observer)
  - Comet Chris (Comet Hunter)
  - Star Mapper Mia (Star Mapper)
  - Rocket Ron (Rocketry Enthusiast)
  - Planet Paige (Planetary Scientist)
  - Galaxy Greg (Galaxy Explorer)
  - Aurora Amy (Aurora Chaser)
  - Meteor Max (Meteor Watcher)
  - Solar Sam (Solar Observer)
- Upcoming Space Events:** A list of events with "Remind Me" buttons:
  - Delta Aquariids Meteor Shower** (Jul 21): Peak viewing for this annual meteor shower.
  - Perseid Meteor Shower** (Aug 12): One of the best meteor showers of the year.
  - Neptune at Opposition** (Sep 18): Neptune is closest to Earth and fully illuminated.

- Comment creation for posts



Stellar Sam  
1h ago

...  
Captured the Milky Way core from a dark sky site. The clarity tonight was out of this world! 🌌

1 2 2

Luna Lee Amazing shot!

Comet Chris What camera did you use?

Wow AweZsome!!!

Post

- Comment Edit Option

 **Stellar Sam**  
1h ago ...



Captured the Milky Way core from a dark sky site. The clarity tonight was out of this world! 🌌

   3  2

 **Luna Lee** Amazing shot!

 **Comet Chris** What camera did you use?

 **You** Wow AweZome!!!  Cancel

- Comment Deletion Option

Nebula Nate  
2h ago



First light with my new telescope! Managed to spot Saturn and its rings. Absolutely mesmerizing. 🌟

1 2 3

Rocket Ron Saturn is my favorite!

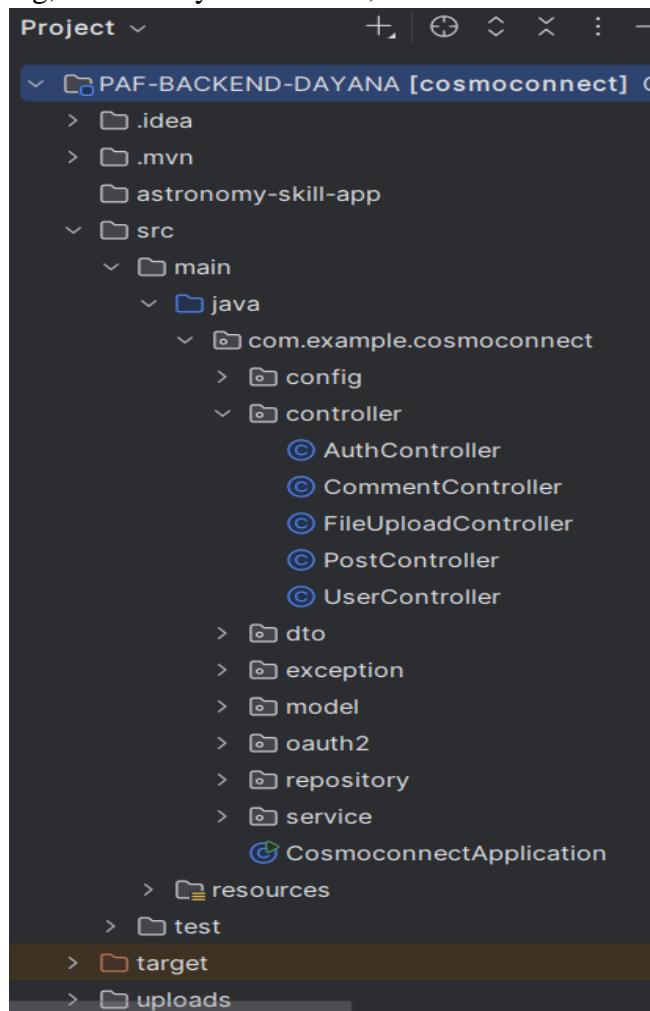
You Awesome shot!! ✎

Add a comment... Post

## Backend Implementation

The backend of the *CosmoConnect* application is developed using Spring Boot and follows a modular architecture with well-defined layers. The application is structured into key packages such as controller, service, repository, model, and dto, ensuring clean separation of responsibilities. The PostController manages all endpoints related to post creation, retrieval, updating, deletion, and interaction features like like and unlike. The endpoints are RESTful, with support for POST, GET, PUT, PATCH, and DELETE methods. Actions such as uploading media files, tagging mentors, and managing collaborative posts are handled by the PostService layer, which contains the core business logic. Data is transferred securely between the frontend and backend using PostDto, and persisted using PostRepository with MongoDB as the database. The Post model defines attributes such as title, description, media, tags, and user references.

Similarly, the CommentController manages endpoints for creating, reading, updating, and deleting comments. Comments are linked to posts using the CommentService, and the data structure is defined in the Comment model. Secure data transfer is facilitated by CommentDto, and comment records are stored via CommentRepository. JWT-based authentication ensures that only authorized users can perform actions, with authentication handled through AuthController, UserController, and configurations in the oauth2 and config packages. File uploads for posts are managed by FileUploadController, with file size limits and storage paths defined in application.properties. The application runs on MongoDB (cosmodb) with SQL autoconfiguration disabled for optimized performance. Overall, the backend is designed to be robust, scalable, and secure, supporting dynamic post sharing, community interactions, and real-time user engagement.



# Database Implementation

## Posts Collection

The posts collection in cosmodb stores user-generated content including titles, descriptions, media URLs, telescope specs, observation details, tags, and coordinates. Each post references the author via a DBRef to the users collection. It also tracks likes, visibility (public/private), and supports rich metadata like exposure time and equipment details.

The screenshot shows the MongoDB Compass interface connected to localhost:27017. The left sidebar displays the database structure under 'Connections (1)'. The 'posts' collection is selected under the 'cosmodb' database. The main pane shows two documents in the 'Documents' tab. The first document is a detailed astronomical observation:

```
_id: ObjectId('680a8c4503109e7669f002df')
title: "My First Astronomical Observation"
description: "Observing the Andromeda Galaxy"
content: "Detailed observation notes..."
mediaUrls: Array (1)
mediaType: "IMAGE"
telescopeUsed: "Celestron NexStar 8SE"
location: "Backyard Observatory"
celestialObject: "Andromeda Galaxy"
observationDateTime: 2024-03-20T14:30:00.000+00:00
observationConditions: "Clear skies, low light pollution"
author: DBRef('users', '6807a22075c4196e8602c0b8')
createdAt: 2025-04-24T19:08:53.878+00:00
updatedAt: 2025-04-24T19:08:53.878+00:00
likes: Array (empty)
comments: Array (empty)
isPublic: false
coordinates: "00h 42m 44.3s +41° 16' 9\"/>
exposureTime: "30 seconds"
equipmentDetails: "Celestron NexStar 8SE, Canon EOS 60D"
processingDetails: "Stacked 10 images using DeepSkyStacker"
_class: "com.example.cosmoconnect.model.Post"
```

The second document is a shorter note:

```
_id: ObjectId('680a9180c1793e1aa719b06d')
title: "post2"
description: "Galaxy"
content: "Detailed observation notes..."
```

## Comments Collection

The comments collection stores individual comment entries linked to specific posts and users using DBRef references. Each document includes the comment text, author, creation and update timestamps, and an array for likes. This structure supports threaded interaction and engagement without duplicating post data, ensuring clean referencing and efficient retrieval.

The screenshot shows the MongoDB Compass interface for the 'comments' collection in the 'cosmodb' database. The left sidebar lists connections and databases, with 'localhost:27017' and 'cosmodb' selected. The main area displays four documents in the 'Documents' tab. Each document is represented by a card with its \_id, content, author, post, creation date, update date, likes array, and class information. The first document's content is 'PRAISE GOD'. The second document's content is 'Thank you'.

Document	Content	Author	Post	Created At	Updated At	Likes	Class
1	PRAISE GOD	DBRef('users', '6807a22075c4196e8602c0b8')	DBRef('posts', '680a9180c1793e1aa719b06d')	2025-04-24T19:41:29.236+00:00	2025-04-24T19:41:29.236+00:00	Array (empty)	com.example.cosmoconnect.model.Comment
2	Thank you	DBRef('users', '6807a22075c4196e8602c0b8')	DBRef('posts', '680a9180c1793e1aa719b06d')	2025-04-24T19:44:39.836+00:00	2025-04-24T19:44:39.836+00:00	Array (empty)	com.example.cosmoconnect.model.Comment
3							
4							

# Learning Progress CRUD Functionality - IT22057488

## Function Overview

The Space Learning Journey – Personal Log & Showcase module enables learners to log and manage their learning experiences in a reflective and orderly manner. Users can log details such as subject, learning aims, duration, and skills gained, with the option to mark entries as private or public. This module also supports future learning by capturing next steps, enabling users to share milestones, and receiving AI-based learning recommendations. It promotes individual growth and community learning by supporting progress sharing among respective communities.

## User Roles & Permissions

### ❖ Registered Users :

- Creates, reads, writes, and deletes own learning logs.
- Makes logs private or public.
- Tracks learning progress with phases and time logs.
- Receives AI-based learning suggestions.
- Adds acquired skills and sets next actions.
- Views their dashboard with private logs and public exhibit.
- Publishes public logs to learning communities.

## Functionalities Covered (CRUD Operations)

### Create – What users can create

- Registered users can create personalized learning logs under the Space Learning Journey module by entering key details such as the learning topic, subject, start and end dates, and a summary of what they learned.
- Users can specify the current progress stage and total time spent, helping them track their learning journey more accurately.
- Users can list multiple skills acquired during the learning process using an array format.
- Users can outline their next learning steps and decide whether the log should be marked as private or public.
- Each entry can contribute to a larger showcase or remain part of a private reflective journal.
- All this data is transmitted under the POST /api/v1/learning/save path.

```

POST http://localhost:8089/api/v1/learning/save
Params Authorization Headers (9) Body Scripts Settings Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify
1 {
2   "learningTopic": "Spring Boot Development",
3   "startDate": "2025-04-10",
4   "endDate": "2025-05-10",
5   "learningSubject": "Software Development",
6   "whatDidYouLearn": "Building RESTful APIs with Spring Boot and MongoDB",
7   "currentProgressStage": 3,
8   "timeSpentInHours": 12.5,
9   "isPublic": true,
10  "skills": ["PHP", "Spring Boot", "MongoDB", "REST API", "Maven"],
11  "nextSteps": "Add authentication and authorization"
12 }

```

Body Cookies Headers (8) Test Results 200 OK 395 ms 277 B Save Response ...

Raw Preview Visualize 1 6819d7ab05fb23efc4361a5

## Read – What Data is Viewable and by whom

- Registered user can view his own learning logs, whether they are public or private.
- Public logs can be accessed by all registered users for sharing information, community discussion, or personal inspiration.
- Community sections or user profiles can also show public logs, listed skills, and milestone entries in the showcase view.
- The endpoints applicable for reading data are GET /api/v1/learning/getAll to read all records and GET /api/v1/learning/learning/{id} to read a specific log by its ID.

```

GET http://localhost:8089/api/v1/learning/getAll
Params Authorization Headers (6) Body Scripts Settings Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify
1

```

Body Cookies Headers (8) Test Results 200 OK 114 ms 1.09 KB Save Response ...

{ } JSON Preview Visualize 1 [
2 {
3 "id": "680c95fcb0fee26d5618b832",
4 "learningTopic": "GraphQL Architecture",
5 "startDate": "2025-03-15",
6 "endDate": "2025-04-27",
7 "learningSubject": "Web Development",
8 "whatDidYouLearn": "GraphQL servers with Apollo and integrating with React frontend",
9 "currentProgressStage": 2,
10 "timeSpentInHours": 15.7,
11 "skills": [
12 "Schema design",
13 "Resolver implementation",
14 "Error handling"
15 ],
16 "nextSteps": "Implement authentication middleware and subscription functionality",
17 "public": false
18 },
19 ]

Space Learning - Backend / Read by ID

GET http://localhost:8089/api/v1/learning/learning/680c95fcb0fee26d5618b832

Params Authorization Headers (6) Body Scripts Settings Cookies

form-data x-www-form-urlencoded raw binary GraphQL JSON

1

Body Cookies Headers (8) Test Results

200 OK 11 ms 707 B Save Response

{ } JSON Preview Visualize

```
1 {  
2   "id": "680c95fcb0fee26d5618b832",  
3   "learningTopic": "GraphQL Architecture",  
4   "startDate": "2025-03-15",  
5   "endDate": "2025-04-27",  
6   "learningSubject": "Web Development",  
7   "whatDidYouLearn": "GraphQL servers with Apollo and integrating with React frontend",  
8   "currentProgressStage": 2,  
9   "timeSpentInHours": 15.7,  
10  "skills": [  
11    "Schema design",  
12    "Resolver implementation",  
13    "Error handling"  
14  ],  
15  "nextSteps": "Implement authentication middleware and subscription functionality",  
16  "public": false  
17 }
```

## **Update – What can be modified and under what conditions**

- Registered users are also allowed to update all fields of any section of their existing learning logs, the subject, topic, description of learning, start date and end date, progress stage, time spent, and learned skills.
  - They also have the choice of updating the specified next actions and isPublic toggle visibility setting on and off for viewing as private and public.
  - Updates are supported only on the user-owned records where ownership is checked first before actually processing the requested update.
  - Updates are performed through the PUT /api/v1/learning/edit/{id} endpoint.

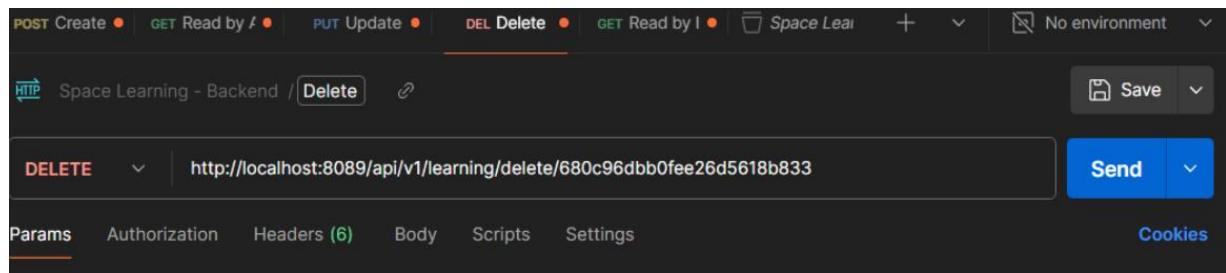
The screenshot shows a POSTman interface with the following details:

- Method:** PUT
- URL:** `http://localhost:8089/api/v1/learning/edit/680c96dbb0fee26d5618b833`
- Body:** Raw JSON payload (selected tab)
- Body Content:**

```
1 {  
2     "learningTopic": "GraphQL Architecture",  
3     "startDate": "2025-03-15",  
4     "endDate": "2025-04-27",  
5     "learningSubject": "Web Development",  
6     "whatDidYouLearn": "GraphQL server with Apollo and integrating with React frontend",  
7     "currentProgressStage": 2,  
8     "timeSpentInHours": 15.7,  
9     "isPublic": true,  
10    "skills": ["Schema design", "Resolver implementation", "Error handling"],  
11    "nextSteps": "Implement authentication middleware and subscription functionality"  
12}
```
- Response Status:** 200 OK
- Response Headers:** 33 ms, 708 B
- Response Body:** The same JSON object as the request body, indicating successful update.

## Delete – What can be removed?

- Registered users can delete only their own learning logs, allowing them to manage and clean up their personal learning history as needed.
- Deletion is strictly restricted to the log creator; users cannot delete logs created by others.
- Each delete request is validated against user ownership before it is processed to ensure secure and authorized access.
- Deletions are handled through the DELETE /api/v1/learning/delete/{id} endpoint.



## User Interface Design (UI Screenshots)

- Learning Progress Creation Form page

The screenshot shows the 'Log New Learning Progress' form overlaid on the Astro Explorer profile page. The form fields include:

- Learning Topic:** Telescope Basics
- Learning Subject:** Introduction to beginners
- What Did You Learn?** Learned about telescope types, magnification, and proper setup techniques. Explored different eyepieces and their uses for various celestial observations.
- Date of Progress:** Start Date: 05/07/2025, End Date: 05/24/2025
- Skills Gained:** Telescope Setup, Polar Alignment
- Suggested skills for this topic:** Telescope Setup, Mirror Alignment, Focusing, Polar Alignment
- Next Steps:** Align your telescope for tonight's sky, Try observing the Moon's craters, Experiment with different eyepieces
- Buttons:** Launch Post, Add, Log Progress

- Learning Progress Read Page

The screenshot shows the COSMIC app interface. At the top, there's a navigation bar with icons for Feed, Learn, Compete, Explore, and a user profile icon. The main area has a dark background with purple and white text.

**Profile Section:**

- Profile Picture:** Placeholder for a profile picture, labeled "Profile".
- Name:** Astro Explorer
- Bio:** Passionate about exploring the cosmos and sharing astronomical knowledge. Currently learning about deep space objects and astrophotography.
- Learning Progress:** 75% (represented by a progress bar)
- Followers:** 128
- Following:** 64
- Skills:** Telescope Operation, Astrophotography, Star Mapping

**Share Section:**

- Profile:** Share your cosmic discoveries...
- Icons:** Sun, Moon, Star
- Buttons:** Launch Post

**Progress Section:**

- Progress Tab:** Selected tab (purple background)
- Community Tab:** Unselected tab (grey background)
- Skills:** Cosmic Photography, Telescope Basics
- Cosmic Photography Card:** Introduction of Cosmic, 5/7/2025, 10 hrs, You're orbiting greatness, Click to Flip
- Telescope Basics Card:** Introduction to beginners, 5/7/2025, 7.5 hrs, Steady like the moon, Click to Flip

- Learning Progress Flip Card UI page

The screenshot displays the COSMIC Learning Progress Flip Card UI page. At the top, there's a navigation bar with icons for Feed, Learn, Compete, Explore, and a user profile icon. The main area features a profile card on the left and a progress card on the right.

**Profile Card (Left):**

- Profile:** Shows a placeholder circular profile picture.
- Astro Explorer:** User title.
- Description:** Passionate about exploring the cosmos and sharing astronomical knowledge. Currently learning about deep space objects and astrophotography.
- Learning Progress:** A progress bar at 75% completion.
- Followers:** 128 Followers.
- Following:** 64 Following.
- Skills:** Telescope Operation, Astrophotography, Star Mapping.

**Progress Card (Right):**

- Share your cosmic discoveries...** Input field with three icons below it: a globe, a crescent moon, and a star.
- Launch Post** button.
- Navigation tabs:** Posts (gray), Progress (purple), Community.
- Progress:** Section header with a plus sign (+).
- Card Example:** A flip card titled "Cosmic Photography" showing an introduction, date (5/7/2025), time (10 hrs), and a message "You're orbiting greatness". It includes a "Click to Flip" button at the bottom.
- Details on the right side of the card:**
  - What Did You Learn?**: Learned about telescope types, magnification, and proper setup techniques. Explored different eyepieces and their uses for various celestial observations.
  - Skills Gained:** Telescope Setup, Polar Alignment.
  - Next Steps:** Align your telescope for tonight's sky, Try observing the Moon's craters, Experiment with different eyepieces.
  - Private** toggle switch.
  - Action buttons:** Edit (pencil) and Delete (trash can).

- Learning Plan Edit Form Page

The screenshot shows the 'Edit Learning Progress' form on the COSMIC platform. The user profile on the left indicates they are an 'Astro Explorer' with 128 Followers and 64 Following. The main form details the learning topic as 'Telescope Basics' and the subject as 'Introduction to beginners'. The 'What Did You Learn?' section contains a text box with the following content:

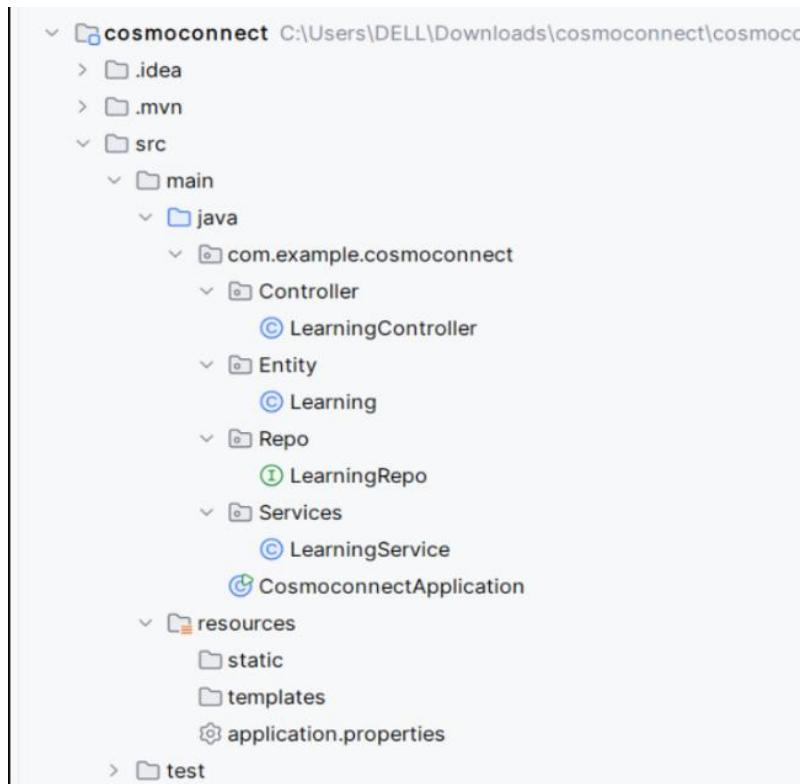
Learned about telescope types, magnification, and proper setup techniques.  
Explored different eyepieces and their uses for various celestial observations.

The 'Date of Progress' section shows the start date as 05/07/2025 and the end date as 05/24/2025. The 'Skills Gained' section lists 'Telescope Setup' and 'Polar Alignment' as learned skills, with an option to add more. Suggested skills include 'Telescope Setup', 'Mirror Alignment', 'Focusing', and 'Polar Alignment'. The 'Next Steps' section lists three actions: 'Align your telescope for tonight's sky', 'Try observing the Moon's craters', and 'Experiment with different eyepieces'. A purple 'Update Progress' button is at the bottom right of the form.

## Backend Implementation

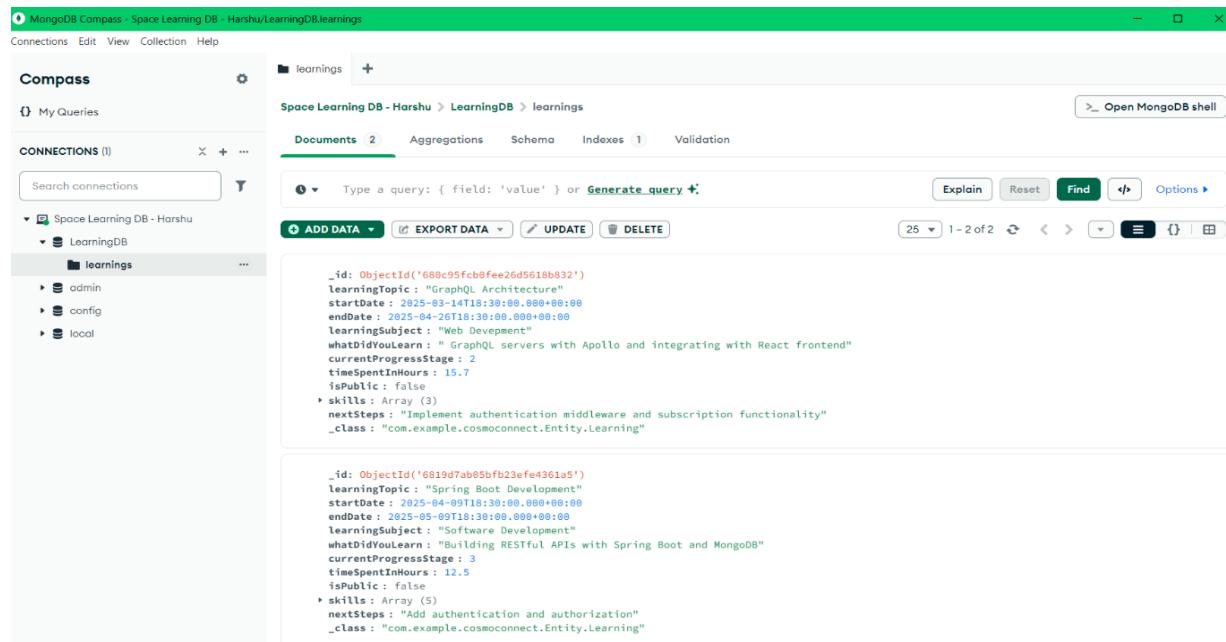
The backend of the Space Learning Journey module is designed on the Spring Boot MVC (Model-View-Controller) architecture, with each layer dedicated to a certain task. The Controller Layer (`LearningController.java`) receives API requests such as creating, updating, deleting, and retrieving learning logs through endpoints such as `/save`, `/edit`, `/delete`, `/getAll`, and `/learning/{id}`.

The Service Layer (`LearningService.java`) encapsulates the core business logic of the application. These activities range from validating the input, validating null, and that the updating and deleting of the logs are only performed by those who have the right to do so. The Repository Layer (`LearningRepo.java`) uses `MongoRepository <Learning, String>`, enabling direct, unambiguous interaction with the MongoDB database. The Entity class (`Learning.java`) is the structure of a learning log made up of fields like topic, subject, progress stage, skills, time taken, visibility status, and next actions. Each element suits its corresponding position in the MVC pattern so as to have a clean and manageable codebase without file-by-file decomposit.



## Database Implementation

The Space Learning Journey module stores its data in a MongoDB collection named learnings. Each document in this collection is a single learning log that a user has submitted. The structure has fields such as `_id` (an auto-created ObjectId), `learningTopic`, and `learningSubject`, which specify the topic of the entry. Date fields such as `startDate` and `endDate` indicate the learning period. The `whatDidYouLearn` field has a text summary of the knowledge learned, while `currentProgressStage` and `timeSpentInHours` help track the progress of the learner. The `skills` field is an array of technologies or competencies learned. The `nextSteps` field outlines the direction of future learning for the user. Finally, `isPublic` is a Boolean flag that determines whether the log is public or not. Its document-centered, extensible schema supports convenient access and expansion of customized learning reports.



```
_id: ObjectId('680c95fc0fee26d5618b832')
learningTopic: "GraphQL Architecture"
startDate: 2025-03-14T18:38:00.000+00:00
endDate: 2025-04-26T18:38:00.000+00:00
learningSubject: "Web Development"
whatDidYouLearn: "GraphQL servers with Apollo and integrating with React frontend"
currentProgressStage: 2
timeSpentInHours: 15.7
isPublic: false
skills: Array (3)
nextSteps: "Implement authentication middleware and subscription functionality"
_class: "com.example.cosmoconnect.Entity.Learning"

_id: ObjectId('6819d7ab05fb23efe4361a5')
learningTopic: "Spring Boot Development"
startDate: 2025-04-09T18:38:00.000+00:00
endDate: 2025-05-09T18:38:00.000+00:00
learningSubject: "Software Development"
whatDidYouLearn: "Building RESTful APIs with Spring Boot and MongoDB"
currentProgressStage: 3
timeSpentInHours: 12.5
isPublic: false
skills: Array (5)
nextSteps: "Add authentication and authorization"
_class: "com.example.cosmoconnect.Entity.Learning"
```

## Summary

The Learning Journey module is a comprehensive platform that can be utilized to facilitate individual and collaborative learning experiences. It enables users to capture, monitor, and review their learning progress through properly organized CRUD functionalities. Through the use of the flexibility of MongoDB and the reliability of the Spring Boot framework, the module ensures effective data management and scalability. Users own their learning entries fully, including provision to manage visibility and share milestones outside. This supports reflective learning culture with facilitation of community engagement and continuous acquisition of skills.

# Competition Management CRUD Functionality - IT22294098

## Function Overview

The Competition Management module in the Admin Panel allows administrators to efficiently organize and oversee competitions within the platform. It provides tools to create, configure, and manage competitions by setting titles, categories, formats, and descriptions. Admins can also schedule timelines, upload relevant resources, assign judges, and define rules or FAQs. Additionally, the module enables dynamic features such as countdown timers, leaderboards, Q&A sessions, and filtering options. Its primary purpose is to streamline end-to-end competition lifecycle management while enhancing clarity, engagement, and fairness.

## User Roles & Permissions

### ❖ Admin

- Creating new competitions with detailed configurations such as title, category, description, and format.
- Viewing and managing all existing competitions.
- Updating competition details including timelines, rules, FAQs, and judging criteria.
- Uploading and managing supporting materials like banners, documents, datasets, and additional resources.
- Assigning and managing judges for each competition.
- Scheduling and updating Q&A sessions to support participants.
- Enabling or disabling features like countdown timers and public leaderboards.
- Filtering and organizing competitions based on various parameters such as category, level, status, or featured tags.
- Archiving or canceling competitions without permanently deleting them.

### ❖ Registered Users or Participants

- View available competitions.
- Submit entries or participate according to the competition rules.
- Access public content such as timelines, FAQs, and results.
- Track leaderboard rankings and countdowns if made visible by the admin.

## Functionalities Covered (CRUD Operations)

### Create- What Admins can create

- Providing the competition title, category, format, and description.
- Defining timelines: start date, submission deadline, and result announcement.
- Uploading supporting materials such as banners, documents, datasets, and learning resources.

The screenshot shows a POST request to `http://localhost:8080/api/v1/competition/add`. The 'Body' tab is selected, showing form-data parameters:

Key	Type	Value
competitionTitle	Text	Zero Gravity Robotics Competition
competitionCategory	Text	Robotics in Space
competitionType	Text	Team
maxTeamSize	Text	4
competitionDescription	Text	Design an innovative Mars rover prototype that can navigate challenging terrain an...
problemStatement	Text	Create a detailed design for a Mars rover that can operate in extreme temperature...
startDate	Text	2025-05-15
submissionDeadline	Text	2025-06-30
competitionStatus	Text	Active
countdownTimerEnabled	Text	true
competitionBanner	File	space2.jpg
competitionInstructions	File	Space_Innovation_Competition_Guide.pdf

### Read- What data is viewable and by whom?

- View all competition details including filters (category, level, status), assigned judges, uploaded files, and current configurations.
- Access dashboard elements like leaderboards, countdowns, and scheduled Q&A sessions.
- View public competition information such as titles, descriptions, timelines, rules, FAQs, and leaderboards .based on what admins have enabled for visibility.

The screenshot shows a GET request to `http://localhost:8080/api/v1/competition/getAll`. The 'Headers' tab is selected, showing the following headers:

Header	Value
Content-Type	application/json

## Update- What can be modified and under what conditions?

- Modify existing competition details including timelines, descriptions, judging criteria, and resources.
- Reschedule Q&A sessions and update rules or FAQs.
- Adjust visibility settings and toggle featured status for competitions

The screenshot shows the CosmoConnect API tool interface. The URL is `http://localhost:8080/api/v1/competition/edit/680c947091bc7d3c1cb0cd50`. The method is set to `PUT`. The `Body` tab is selected, showing a table of competition parameters and their values:

Key	Type	Value
competitionTitle	Text	Zero Gravity Robotics Competition
competitionCategory	Text	Robotics in Space
competitionType	Text	Team
maxTeamSize	Text	5
competitionDescription	Text	Updated description
problemStatement	Text	Updated problem statement
startDate	Text	2024-03-20
submissionDeadline	Text	2024-03-25
competitionStatus	Text	Active
countdownTimerEnabled	Text	true
competitionBanner	File	space2.jpg
competitionInstructions	File	Space_Innovation_Competition_Guide.pdf

## Delete - What can be removed, and are there any restrictions

- Archive or cancel competitions without permanently deleting data.
- Remove competitions from public visibility while retaining them for internal reference or audit purposes.

The screenshot shows the CosmoConnect API tool interface. The URL is `http://localhost:8080/api/v1/competition/delete/680c574b1f7b943b0f48b371`. The method is set to `DELETE`. The `Send` button is visible at the top right.

## User Interface Design (Screenshots)

- Admin Dashboard with all competitions displayed

The screenshot displays the COSMO CONNECT Admin Control Center. At the top, there are navigation links: Feed, Learn, Compete, Explore, Posts, and a user profile icon. The main header "Admin Control Center" is followed by a subtitle "Manage competitions and monitor activities". On the right, a clock shows "21:07:00" and the date "Wednesday, May 7, 2025". Below this, four summary cards show the count of competitions: Total Competitions (2), Ongoing (1), Upcoming (1), and Completed (0). Two competition cards are listed below: "Astrophotography Challenge 2024" (Upcoming) and "Space DIY Project Competition" (Ongoing). Each card includes a thumbnail, category, description, type, deadline, and three action buttons: View Details, Edit, and Delete. A "Quick Actions" section at the bottom contains three buttons: "Add New Competition" (purple), "View Archived" (blue), and "Manage Judges" (green).

**Admin Control Center**  
Manage competitions and monitor activities

21:07:00  
Wednesday, May 7, 2025

Total Competitions 2

Ongoing 1

Upcoming 1

Completed 0

**Astrophotography Challenge 2024** (Upcoming)  
Capture the beauty of the night sky in this exciting astrophotography competition.  
Category: Astrophotography  
Type: Individual  
Deadline: 4/1/2024  
View Details Edit Delete

**Space DIY Project Competition** (Ongoing)  
Build your own space-related project and showcase your engineering skills.  
Category: Space DIY  
Type: Team  
Deadline: 5/15/2024  
View Details Edit Delete

**Quick Actions**

Add New Competition  
Create a new competition with detailed settings

View Archived  
Access past competitions and their results

Manage Judges  
Add or remove competition judges

- Competiton Creation Page (Form)

**Add New Competition**

**Title:** [Input field]

**Category:** Select Category [Dropdown menu]

**Description:** [Text area]

**Problem Statement:** [Text area]

**Competition Type:** Individual [Dropdown menu]

**Status:** Upcoming [Dropdown menu]

**Start Date:** dd/mm/yyyy [Input field]

**Submission Deadline:** dd/mm/yyyy [Input field]

**Result Announcement Date:** dd/mm/yyyy [Input field]

**Create Competition** [Purple button]

- Competition displayed in the user interface

**Competitions**

**Astrophotography Challenge 2024** [Upcoming]

Capture the beauty of the night sky in this exciting astrophotography competition.

Category: Astrophotography  
Type: Individual  
Deadline: 4/1/2024

**View Details** [Purple button] **Edit** [Blue button] **Delete** [Red button]

**Space DIY Project Competition** [Ongoing]

Build your own space-related project and showcase your engineering skills.

Category: Space DIY  
Type: Team  
Deadline: 5/15/2024

**View Details** [Purple button] **Edit** [Blue button] **Delete** [Red button]

- Competition displayed in detail

The screenshot shows a competition page titled "Space DIY Project Competition" on the COSMO CONNECT platform. The page includes sections for Description, Problem Statement, Timeline, Rules and Guidelines, and a "Submit Entry" button. The timeline shows a start date of 2/15/2024, 12:00:00 AM, a submission deadline of 5/15/2024, 11:59:59 PM, and a result announcement date of 6/1/2024, 12:00:00 AM.

## Backend Implementation

The backend of the Competition Management module is built using a structured Spring Boot architecture, organized into four key layers: Controller, Service, Repository, and Model. At the Controller layer, the `CompetitionController` class is responsible for managing all REST API endpoints related to CRUD operations. These include endpoints for creating new competitions, retrieving existing competitions, updating details, and logically deleting (archiving or canceling) competitions.

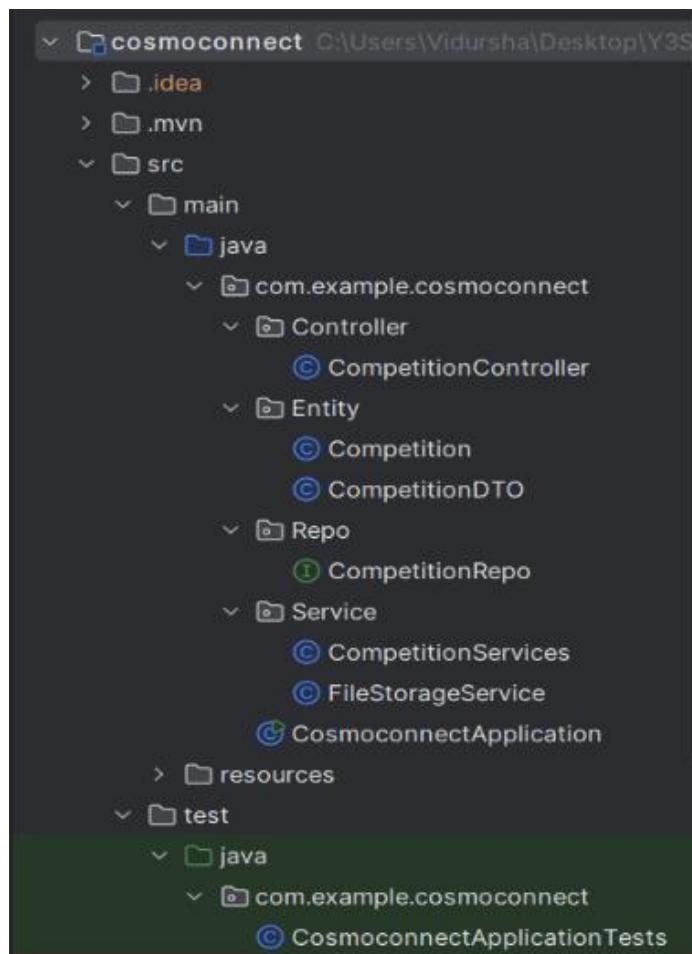
The Service layer is implemented through the `CompetitionServices` class, which contains the core business logic. This includes validating important constraints such as ensuring that the result announcement date occurs after the submission deadline. The service layer also manages conditional updates to fields like FAQs or judging rules, and coordinates file handling for uploaded resources via the `FileStorageService`.

In the Repository layer, the `CompetitionRepo` interface extends `MongoRepository`, enabling seamless interaction with the MongoDB database. This allows for efficient execution of all standard CRUD operations and direct communication with the competitions collection.

The Model layer comprises two main components: the Competition class and the CompetitionDTO. The Competition class defines the structure of the competition entity, including fields such as title, category, timelines, judges, and leaderboard status. The CompetitionDTO is used to transfer validated data between the client and server, ensuring data integrity and consistency throughout the application.

The module exposes well-defined REST API endpoints following standard conventions:

- POST /api/v1/competition/add is used to create a new competition,
- GET /api/v1/competition/getAll retrieves all competitions,
- PUT /api/v1/competition/edit/{id} updates a specific competition, and
- DELETE /api/v1/competition/delete/{id} archives or cancels a competition entry without permanently deleting it.



## Database Structure

The backend of the Competition Management module utilizes MongoDB as the primary database, with all competition records stored in the competitions collection under the CCCompetitionDB database. Each document in this collection represents a single competition and contains all relevant information required for its creation, management, and display.

Key fields in each document include the competitionTitle, competitionCategory, and competitionType (e.g., Individual or Team), along with the maxTeamSize to define team limits. The competitionDescription and problemStatement provide context and explain the challenge to participants. Important timeline fields such as startDate and submissionDeadline are also stored to manage scheduling and countdown functionalities.

Additional fields include competitionStatus to indicate whether the competition is active or archived, and countdownTimerEnabled, which controls the visibility of real-time deadline tracking. Uploaded files such as competition\_instructions (PDFs) and competitionBanner (images) are stored as file names or paths, which are used for participant guidance and visual representation. This document structure is mapped to the Competition model class in the backend and managed through the CompetitionRepo interface for smooth CRUD operations. The flexible schema design supports both current functionality and future enhancements such as scoring, judge assignments, and participant submissions, ensuring scalability and maintainability for the CosmoConnect platform.

The screenshot shows the Compass MongoDB interface. The top navigation bar includes 'Connections', 'Edit', 'View', 'Collection', and 'Help'. Below the navigation is a 'Welcome' banner for 'competitions' on 'localhost:27017 > CCCompetitionDB > competitions'. The main area has tabs for 'Documents' (4), 'Aggregations', 'Schema', 'Indexes' (1), and 'Validation'. A search bar says 'Type a query: { field: 'value' } or Generate query'. Below it are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. A table lists four documents with their \_id, competitionTitle, competitionCategory, competitionType, maxTeamSize, competitionDescription, problemStatement, startDate, submissionDeadline, competitionStatus, countdownTimerEnabled, competition\_instructions, competitionBanner, and \_class. The first document's details are expanded:

```
_id: ObjectId('680a5f1dc45a2a69dbdc724e')
competitionTitle : "Updated Mars Rover Design Challenge"
competitionCategory : "Engineering"
competitionType : "Team"
maxTeamSize : 5
competitionDescription : "Updated description"
problemStatement : "Updated problem statement"
startDate : 2024-03-19T18:30:00.000+00:00
submissionDeadline : 2024-03-24T18:30:00.000+00:00
competitionStatus : "Active"
countdownTimerEnabled : true
competition_instructions : "c4fbcl37-64b3-47eb-b335-9b5681f6bde6.pdf"
competitionBanner : "c5c5c62c-32b1-426e-9e2f-1cfb21ea183c.jpg"
_class : "com.example.cosmocconnect.Entity.Competition"
```

The left sidebar shows 'CONNECTIONS (3)' with entries for localhost:27017 (selected), localhost:27018, localhost:27019, and localhost:27020. It also lists databases: CCCompetitionDB (selected), CompetitionsDb, admin, architecture, config, construction, cosmodb, and local.

# Learning Plan CRUD Functionality - IT22289384

## Function Overview

The Galactic Learning Roadmap module enables users to build and share personalized space learning plans that are organized into clear phases, topics, goals, timelines, and recommended resources. It helps learners structure their educational path using a mission-style layout and enhance each stage with materials like PDFs, toolkits, assessments, and external links. Users can customize their plans, monitor their learning milestones, and receive recognition through badges and certificates. This module also promotes collaborative learning by allowing peer or mentor contributions and supports user motivation through AI-powered boosters and public showcases. It aims to provide a structured, inspiring, and engaging space learning experience tailored to individual growth.

## User Roles & Permissions

### ❖ Guest User:

- Can view public learning plans shared by registered users.
- Can view embedded learning plan previews in posts or public community pages.
- Cannot create, edit, or delete learning plans.
- Cannot participate in assessments, receive certificates, or access personalized boosters.
- Must register to collaborate on plans or save copies for personal use.

### ❖ Mentor:

- Can be invited by registered users to collaborate on specific learning plans.
- Can edit learning plans they have been granted access to by the plan owner.
- Can contribute curated resources, assessments, and feedback within shared learning plans.
- Can help manage visibility settings and milestone tracking for the plans they co-manage.
- Cannot create their own learning plans unless registered as a full user.
- Can assist in enhancing learning journeys through guidance, but cannot access admin dashboards or approval features.

## Functionalities Covered (CRUD Operations)

### Create – What can users create?

- Users can create personalized astronomy learning plans that include structured phases, space-related topics, objectives, timelines, and attached resources such as PDFs, toolkits, and external links.
- Users can create quizzes and assessments within each learning plan and provide summary overviews.
- Users can generate mission-style visual layouts and define badge milestones.
- Users can initiate shared plans by inviting peers or mentors to collaborate.
- All this information is sent through the POST endpoint at /api/v1/learningplan/save.

The screenshot shows a POST request to `http://localhost:8081/api/v1/learningplan/save`. The request body is a JSON object:

```
1 {
2   "planTitle": "Advanced Astrophysics",
3   "description": "Dive deep into the mysteries of the universe. Study black holes, dark matter, and
4                 the Big Bang theory. Includes mathematical modeling and observational techniques.",
5   "difficultyLevel": "Advanced",
6   "duration": "12 weeks",
7   "planMaterials": [],
8   "certificate": "Advanced Astrophysics Certificate",
9   "imageUrls": []
}
```

The screenshot shows a POST request to `http://localhost:8081/api/v1/learningplan/upload/material/680c819757e2060fd71a5aba`. The request body is a form-data object with a file key:

Key	Value	Description	...	Bulk Edit
file	⚠ Primary-Checkpoint-Englis... <input type="button" value="Upload"/>			
Key	Text	Value	Description	

## Read – What data is viewable and by whom?

- Users can view their own learning plans, whether private or shared.
- Registered users can access public learning plans shared by others for inspiration or duplication.
- Admins can view all learning plans, including private ones, for moderation and platform oversight.
- Shared plans, visible badges, and milestone highlights can be viewed in user profiles and community sections.

The screenshot shows a Postman interface with the following details:

- Method:** GET
- URL:** http://localhost:8081/api/v1/learningplan/getAll
- Headers:** (6) - Authorization, Headers (8), Test Results, etc.
- Body:** raw JSON response (Beautified)
- Response Status:** 200 OK
- Response Time:** 338 ms
- Response Size:** 1.63 KB
- Content:** A JSON array of learning plans, each with fields like \_id, planTitle, description, difficultyLevel, duration, planMaterials, certificate, and imageUrl. One plan is explicitly named "Solar System".

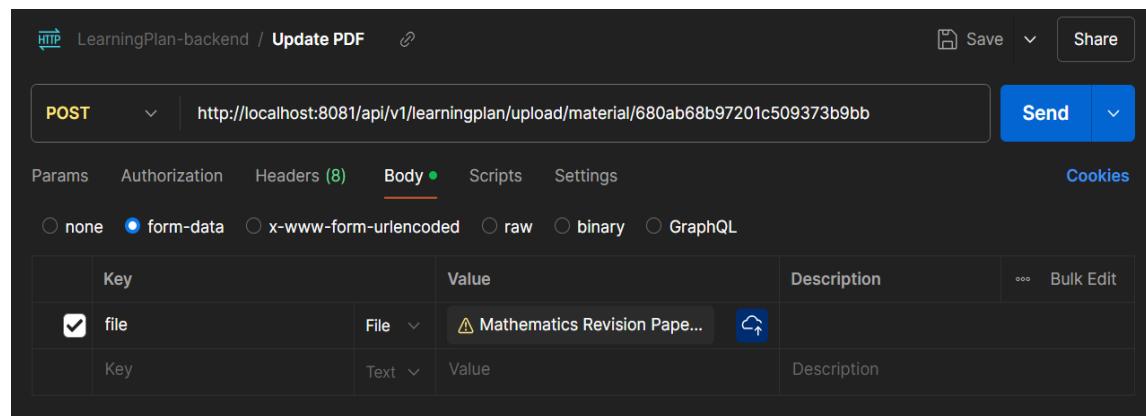
```
1 [  
2 {  
3   "_id": "6801f5922edf1b0551f2f0aa",  
4   "planTitle": null,  
5   "description": null,  
6   "difficultyLevel": null,  
7   "duration": null,  
8   "planMaterials": null,  
9   "certificate": null,  
10  "imageUrls": null  
11 },  
12 {  
13   "_id": "680c7c8157e2060fd71a5ab8",  
14   "planTitle": "Solar System",  
15   "description": "Journey through our cosmic neighborhood..",  
]
```

## Update – What can be modified and under what conditions?

- Users can edit the title, structure, topics, goals, timelines, and resources of their learning plans.
- Users can change the visibility of their plans between private and shared at any time.
- Users can update badge settings, assessment content, and manage collaboration permissions.

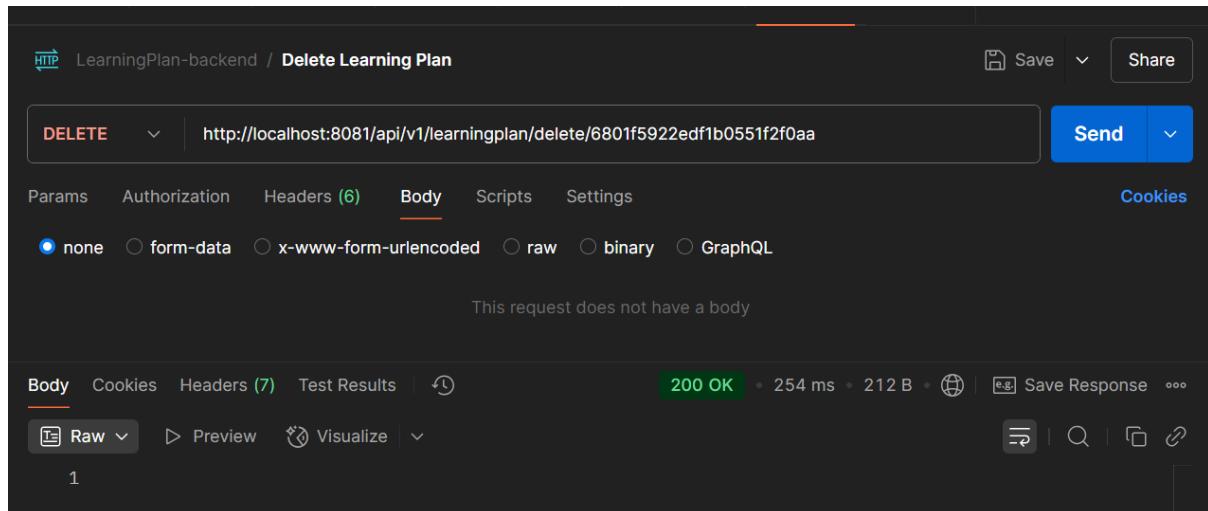
The screenshot shows a POST request in Postman to the URL `http://localhost:8081/api/v1/learningplan/upload/material/680ab68b97201c509373b9bb`. The request body is a JSON object:

```
1 {
2     "planTitle": " Astrophysics",
3     "description": "Dive deep into the mysteries of the universe. Study black holes, dark matter, and
4                     the Big Bang theory. Includes mathematical modeling and observational techniques.",
5     "difficultyLevel": "Advanced",
6     "duration": "12 weeks",
7     "planMaterials": [],
8     "certificate": "Advanced Astrophysics Certificate",
9     "imageUrls": []
}
```



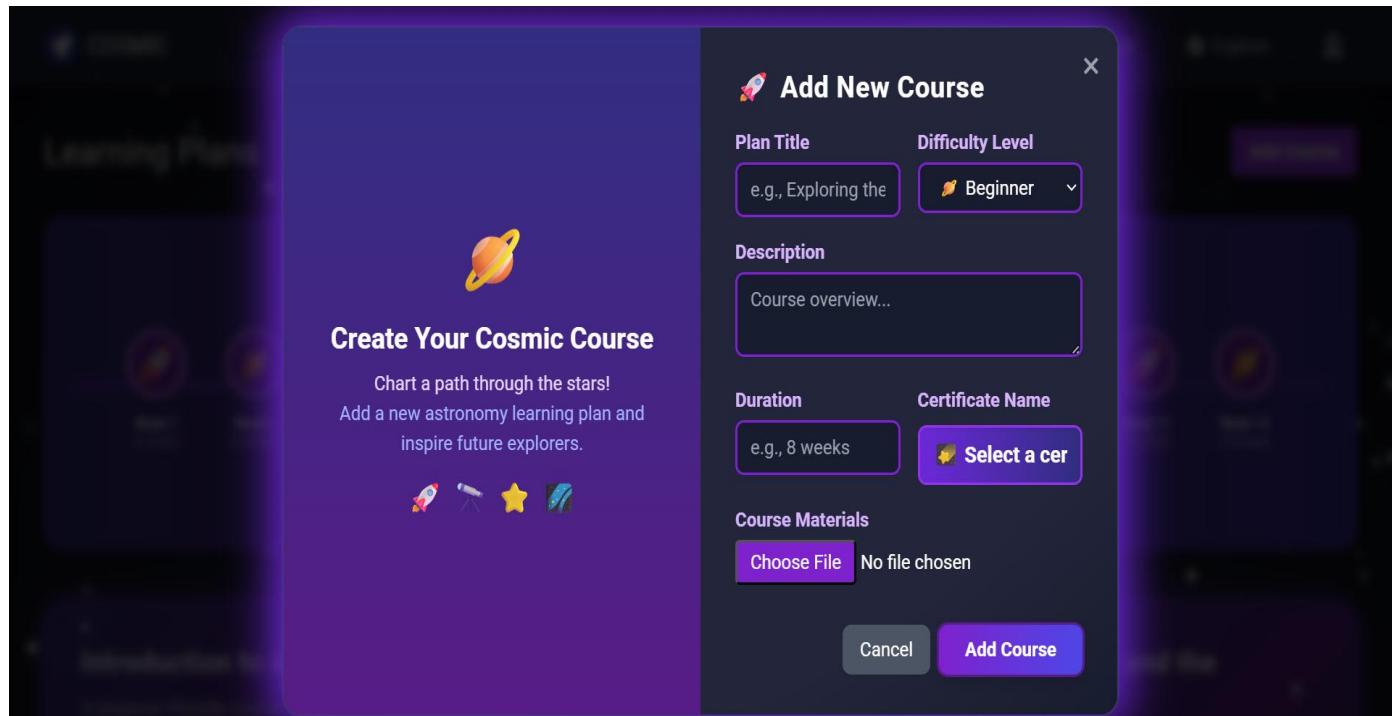
## Delete – What can be removed, and are there any restrictions?

- Users can delete an entire learning plan or remove specific components like topics, quizzes, or resource attachments.
- Admins can delete learning plans that violate platform guidelines.
- Deleted content is permanently removed from public view and platform metrics.



## User Interface Design (Screenshots)

- Learning Plan Creation Page



- Learning Plan Edit Form Page

**Edit Course**

**Plan Title:** Introduction to Astronomy

**Difficulty Level:** Beginner

**Description:** A beginner-friendly course covering the basics of astronomy, celestial objects,

**Duration:** 8 weeks

**Certificate Name:** Astronomy

**Course Materials:** Choose File

**Certificate Options:**

- Select a certificate
- Astronomy Fundamentals Certificate
- Stellar Physics Specialist** (highlighted)
- Cosmology Studies Certificate
- Space Exploration Certificate
- Astrobiology Achievement Certificate

**Cancel**

- Learning Plan Read / Display Page

**COSMIC**

Feed Learn Compete Explore User

**Learning Plans**

**Add Course**

**Introduction to Astronomy Roadmap**

Week	Date	Icon
Week 1	5/7/2025	Rocket
Week 2	5/14/2025	Comet
Week 3	5/21/2025	Sun
Week 4	5/28/2025	Telescope
Week 5	6/4/2025	Alien
<b>Week 6</b>	<b>6/11/2025</b>	<b>Galaxy</b>
Week 7	6/18/2025	Telescope
Week 8	6/25/2025	Alien

**Introduction to Astronomy**

A beginner-friendly course covering the basics of astronomy, celestial objects, and observational techniques.

Difficulty: Beginner  
Duration: 8 weeks  
Certificate: Astronomy Fundamentals Certificate

**Edit** **Delete**

**Advanced Stellar Physics**

Deep dive into stellar evolution, nuclear fusion, and the life cycle of stars.

Difficulty: Advanced  
Duration: 12 weeks  
Certificate: Stellar Physics Specialist

**Edit** **Delete**

**Cosmology and the Universe**

Explore the origins of the universe, dark matter, and the expanding cosmos.

Difficulty: Intermediate  
Duration: 10 weeks  
Certificate: Cosmology Studies Certificate

**Edit** **Delete**

## Backend Implementation

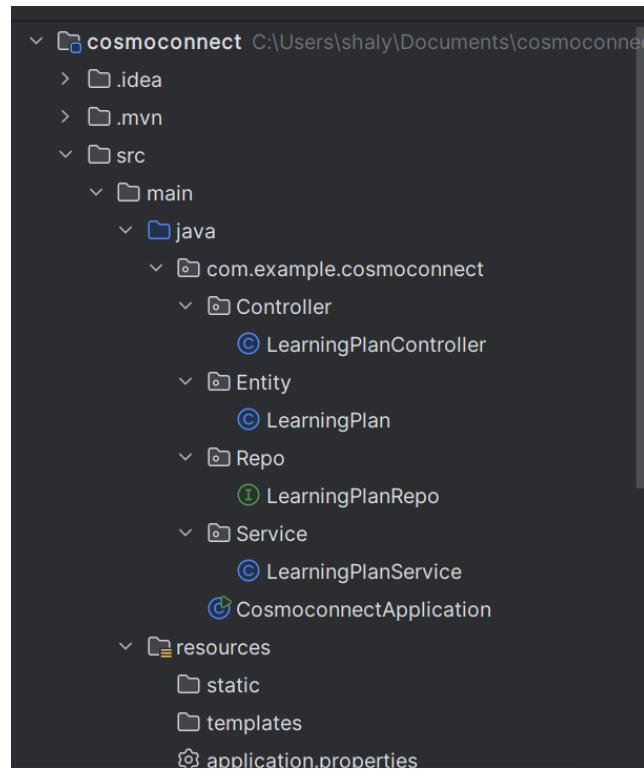
The backend of the Galactic Learning Roadmap module follows the Spring Boot MVC (Model-View-Controller) architectural pattern. The system is modularly organized to support clarity, scalability, and ease of maintenance. The Controller layer (`LearningPlanController.java`) is responsible for handling HTTP requests related to the learning plans. It exposes RESTful endpoints such as `/save`, `/edit/{id}`, `/delete/{id}`, and `/getAll` to perform core CRUD operations. These endpoints act as the entry point for client-side interactions.

The Service layer (`LearningPlanService.java`) contains the core application logic. It manages operations such as creating or updating learning plans, validating inputs, and handling file uploads (e.g., PDFs and images) for each plan. This layer ensures that each uploaded file is stored with a unique identifier and that plan data is updated only when valid.

The Repository layer (`LearningPlanRepo.java`) leverages the `MongoRepository<LearningPlan, String>` interface to provide seamless data access methods, eliminating the need for custom queries for basic operations. This layer serves as a bridge between the service and the database.

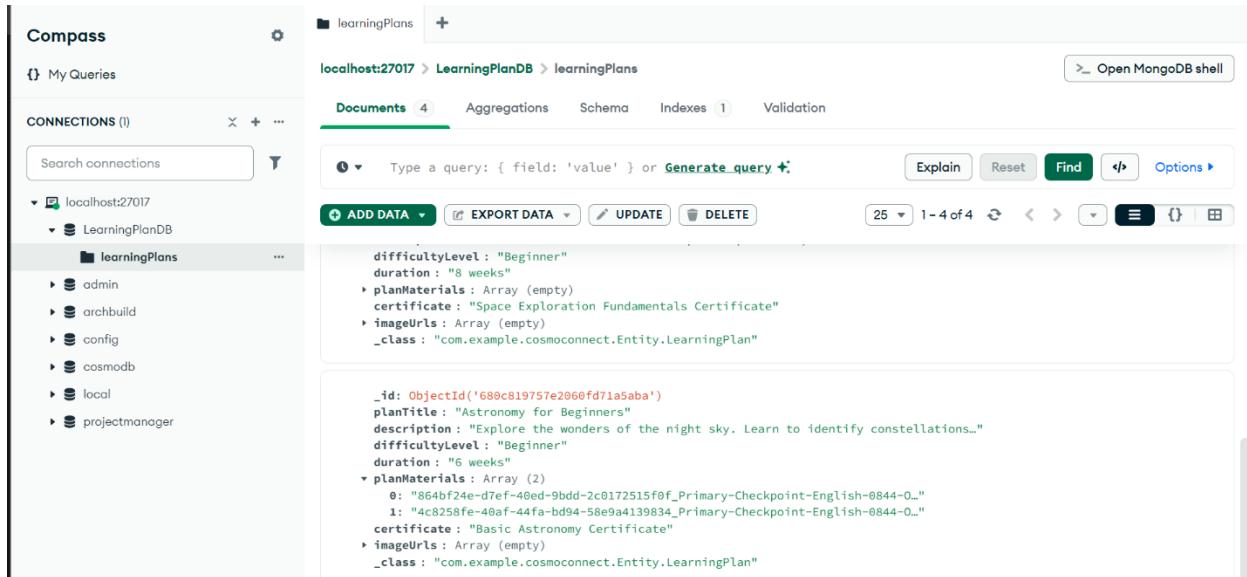
Finally, the Entity class (`LearningPlan.java`) defines the schema for each learning plan. It includes fields such as `planTitle`, `description`, `difficultyLevel`, `duration`, `planMaterials`, `certificate`, and `imageUrls`. Each field corresponds to data captured through the frontend form and persisted in the MongoDB collection named `learningPlans`.

This layered structure supports a clean and maintainable codebase while aligning with industry best practices for Spring Boot applications.



## Database Structure

The Galactic Learning Roadmap module utilizes MongoDB to store user-generated learning plans in a collection named learningPlans, housed within the LearningPlanDB database. Each document captures a single plan and includes fields such as a unique \_id, planTitle, description, difficultyLevel, duration, uploaded planMaterials, and certificate details. Arrays like imageUrls allow media support, while the \_class field maps the document to its corresponding Java entity. The flexible, document-based schema ensures that learning plans are easily stored, retrieved, and updated, with room to expand as new features or fields are introduced.



The screenshot shows the Compass MongoDB interface. On the left, the 'CONNECTIONS' sidebar lists 'localhost:27017' and 'LearningPlanDB'. Under 'LearningPlanDB', the 'learningPlans' collection is selected. The main panel displays the 'Documents' tab with 4 results. A query builder at the top allows typing a query or generating one. Below it are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. The results list two documents:

```
difficultyLevel: "Beginner"
duration: "8 weeks"
planMaterials: Array (empty)
certificate: "Space Exploration Fundamentals Certificate"
imageUrls: Array (empty)
_class: "com.example.cosmocomnnect.Entity.LearningPlan"

_id: ObjectId('680c819757e2060fd71a5aba')
planTitle: "Astronomy for Beginners"
description: "Explore the wonders of the night sky. Learn to identify constellations..."
difficultyLevel: "Beginner"
duration: "6 weeks"
planMaterials: Array (2)
  0: "864bf24e-d7ef-40ed-9bdd-2c0172515f0f_Primary-Checkpoint-English-0844-0_"
  1: "4ac258fe-48af-44fa-bd94-58e9a4139834_Primary-Checkpoint-English-0844-0_"
certificate: "Basic Astronomy Certificate"
imageUrls: Array (empty)
_class: "com.example.cosmocomnnect.Entity.LearningPlan"
```

## Summary

The Galactic Learning Roadmap module delivers a structured and interactive educational experience focused on astronomy. It enables users to design customized learning plans with well-defined topics, phases, timelines, and objectives. Each stage can be enriched with curated resources such as PDFs and toolkits, and visually styled using mission-inspired layouts.

This module facilitates collaborative learning by allowing users to invite peers and mentors to contribute. It introduces gamified progression through badges and certificates, awarded upon reaching milestones. AI-generated learning boosters provide weekly motivation through tailored micro-tasks aligned with user progress.

Overall, the module enhances learner engagement through personalization, collaboration, and timely feedback, making it a valuable addition to a modern space-focused educational platform.

# User Profile Management CRUD functionality – Group Work

## Function Overview

The User Profile Management module allows users to securely register and manage their accounts on the CosmoConnect platform. Users can sign up using traditional email/password or seamlessly authenticate via Google or Facebook OAuth. Each profile contains key details such as full name, email, username, profile picture, and skill tags. The module enables users to update personal information, view others' public profiles, and manage account visibility. Admins maintain full control for moderation and role assignment. This module plays a crucial role in personalizing the user experience and ensuring secure, role-based access across the application.

## User Roles & Permissions

### ❖ Registered Users (Email/Google/Facebook)

- Can register or sign in using email, Google, or Facebook.
- Have a profile automatically created with fetched data (name, email, photo) from OAuth providers.
- Can view and edit their profile (bio, tags, picture).
- Can view public details of other users.

## Functionalities Covered (CRUD Operations)

### Create – What can users create

- Create new profiles using email/password or via **Google/Facebook OAuth**.
- OAuth users have profiles auto-generated on first login with pre-filled fields.
- Optional upload of a custom profile picture after account creation.

### Read – What data is viewable and by whom

- Users can view their own full profile including login method.
- Other users can view public profile details (username, avatar, posts).
- Admins have access to all user details, login source, and account metadata.

### Update – What can be modified and under what conditions

- Users can update their profile picture, bio, skill tags, and display name.
- OAuth users can enhance or modify imported info post-login.
- Admins can edit or correct any profile data, including roles and visibility settings.

### Delete – What can be removed, and are there any restrictions

- Users may request profile deletion (subject to policy).
- Admins can deactivate, soft-delete, or fully remove user accounts.

HTTP <http://localhost:8080/api/auth/login>

POST <http://localhost:8080/api/auth/login> Send

Params Authorization • Headers (9) Body • Scripts Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1  {
2    "email": "test@example.com",
3    "password": "password123"
4  }

```

Beautify

Body Cookies Headers (14) Test Results

200 OK 516 ms 1.24 KB

{ JSON Preview Visualize }

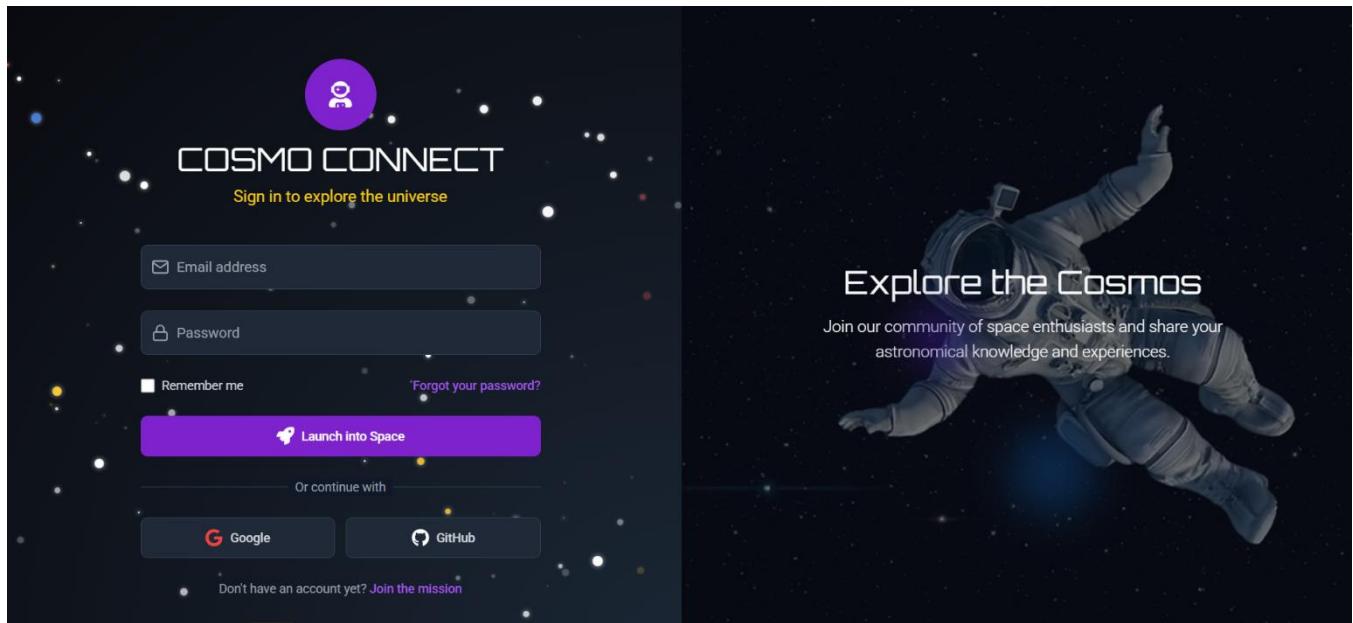
```

1  {
2    "accessToken": "eyJhbGciOiJIUzI1NiJ9.eyJzdWJiOiJ0ZXN0QGV4YW1wbGUuY29tIiwidXNlcm5hbWUiOiJ0ZXN0QGV4YW1wbGUuY29tIiwidXNlcm5hbWUi0iJ0ZXN0QGV4YW1wbGUuY29tIiwidXNlcm5hbWUi0iJ0ZXN0QGV4YW1wbGUuY29tIiwicm9sZXMi0iJST0xFX1VTRViLCJpXXQiOjE3NDY1OTUzOTYsImV4cCI6MTc0NjY4MTc5Nn0.UbgoE_as-_2mcDCOLW-IoTXQ1jw8A1cEbScklWeAykQ",
3    "tokenType": "Bearer",
4    "user": {
5      "id": "6807a22075c4196e8602c0b8",
6      "fullName": "Test User 01",
7      "email": "test@example.com",

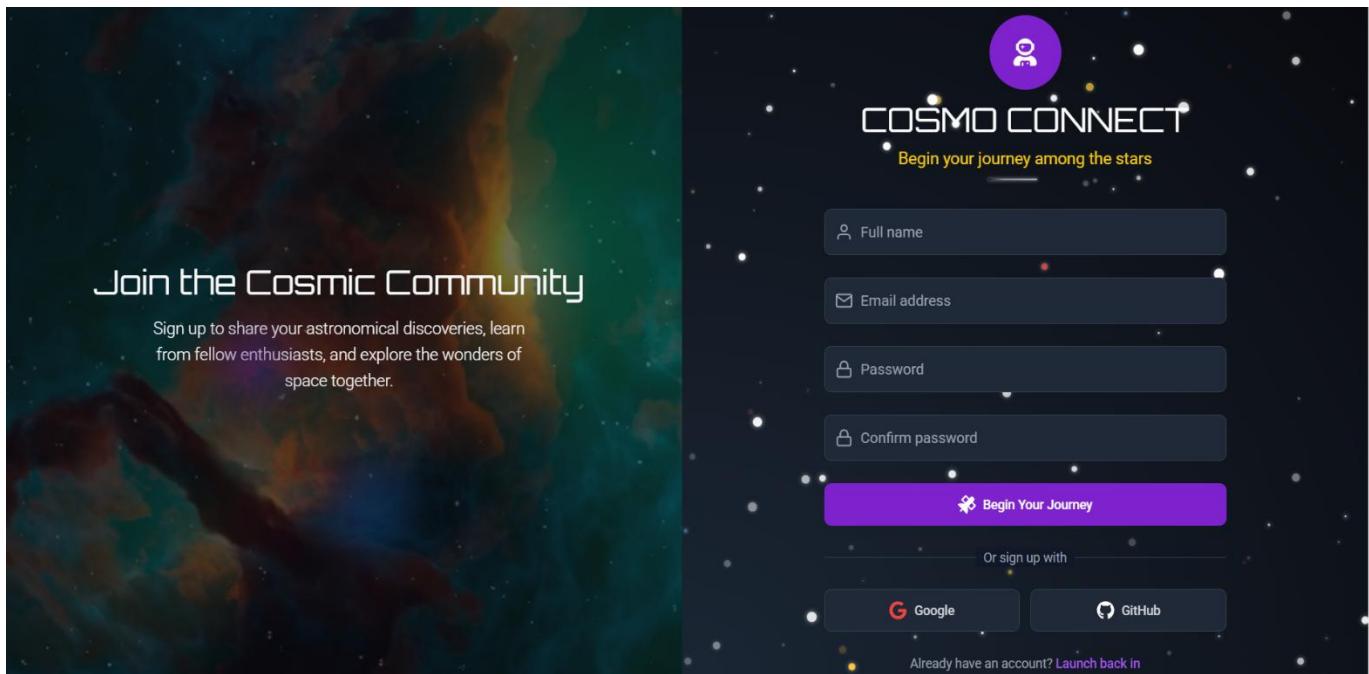
```

## User Interface Design (UI Screenshots)

- Login Page



- **Sign Up Page**



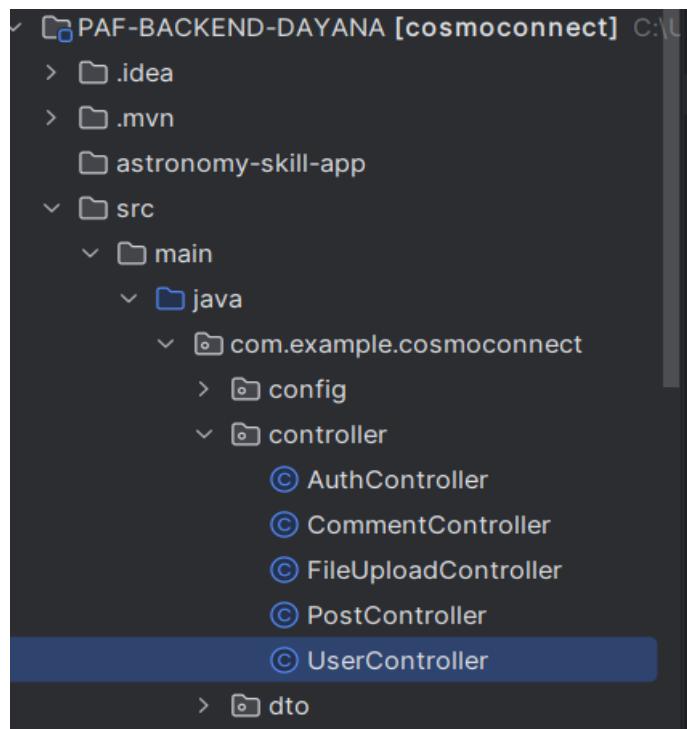
- **User Profile Page**

The user profile page shows a profile picture of a woman named "Astro Explorer". Her bio states she is passionate about exploring the cosmos and learning about deep space objects and astrophotography. She has a learning progress bar at 75%, 128 followers, and 64 following. Under "Skills", she lists "Telescope Operation" and "Astrophotography". On the right, there's a "Feed" section titled "Astronomy Community" featuring profiles for "Astro Explorer 1" through "Astro Explorer 4", each sharing cosmic discoveries. Navigation icons include Feed, Learn, Compete, Explore, View Profile, Settings, and Logout.

## Backend Implementation

The backend for User Profile Management is built using **Spring Boot** and follows a layered architecture, ensuring scalability and security. The UserController handles HTTP requests related to profile creation, retrieval, updating, and deletion. It also manages OAuth login endpoints for both **Google** and **Facebook**, integrating third-party authentication into the system. User registration via email is handled in a dedicated method, while OAuth users are automatically registered upon their first login.

Business logic is implemented in the UserService class, which handles password encryption, email/username validation, profile updates, and role assignments. The UserRepository interacts with MongoDB to persist user data.



## Database Implementation

The **MongoDB** database (cosmodb) stores user profile data in a dedicated users collection. Each document in the collection contains key fields such as username, email, fullName, password (hashed), role (e.g., ADMIN, USER), createdAt, and updatedAt. OAuth users additionally have fields like oauthProvider (Google/Facebook) and oauthId for tracking linked accounts. The profile picture is stored as a URL path, referencing an uploaded file from the server or an external source. User documents also include optional fields like bio, skillTags, and active status. Indexes are created on email and username to ensure uniqueness and speed up lookups. This document-based schema supports flexible expansion and efficient retrieval of user-related data, integrating well with MongoDB's schema-less structure.

The screenshot shows the MongoDB Compass interface connected to the 'cosmodb' database. The 'users' collection is selected, displaying 7 documents. The interface includes a sidebar for connections and collections, a search bar, and various management buttons for adding, updating, and deleting data. Two documents are expanded to show their detailed structure:

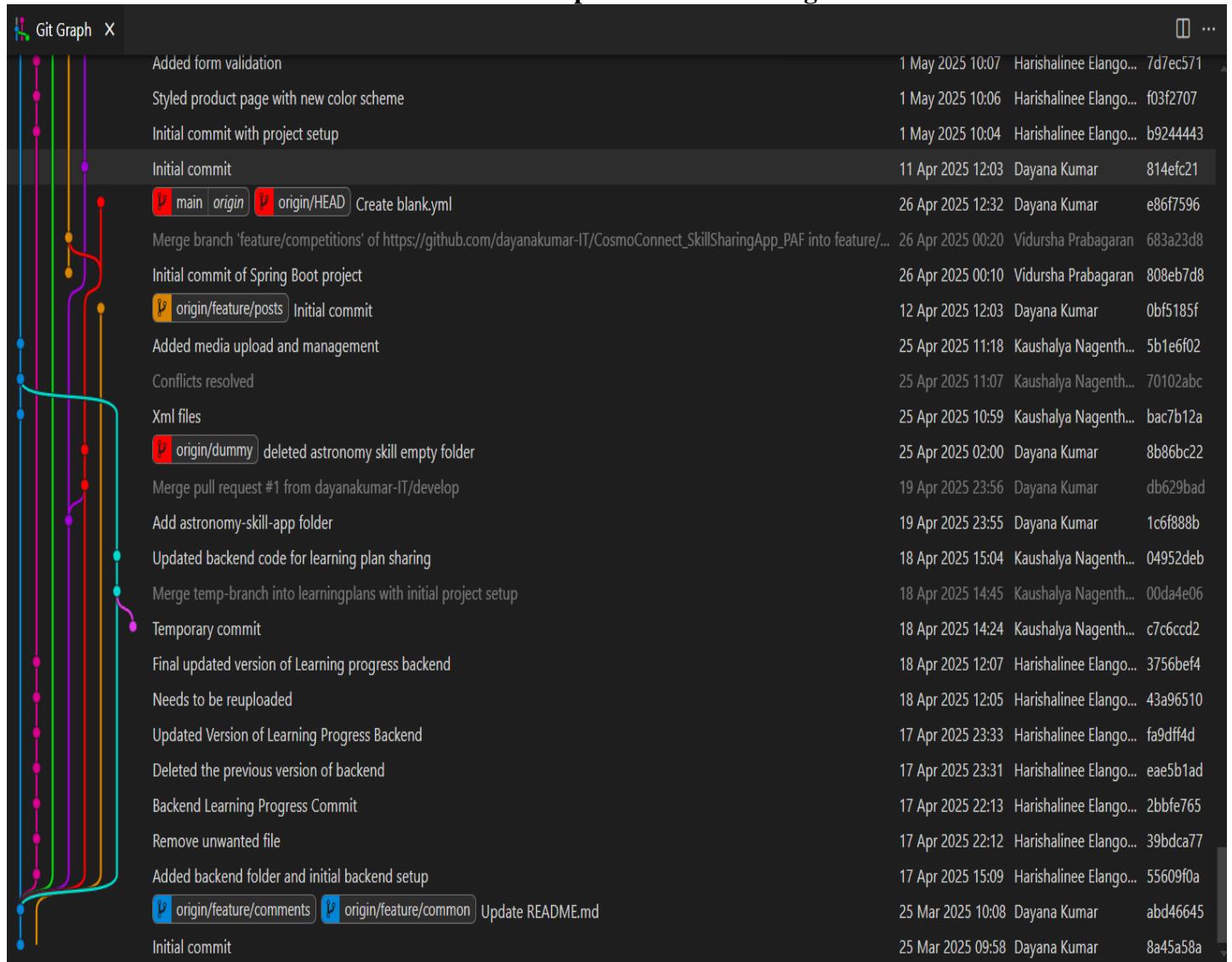
```
_id: ObjectId('6807a22875c4196e8682c0b8')
fullName : "Test User 01"
email : "test@example.com"
password : "$2a$10$WnD1mKo7g/i400N.0vJJ.JFBzu65LetcNH94idPF64pA1PtPfZ"
emailVerified : false
sharePersonalInfo : false
accountCreated : 2025-04-22T14:05:20.207+00:00
roles : Array (1)
enabled : true
provider : "local"
_class : "com.example.cosmocconnect.model.User"

_id: ObjectId('6807a850e888f601475763eb')
fullName : "Test User"
email : "dayana@example.com"
password : "$2a$10$7aYB5NA7bAVTn/lkViwUNOpkSMxrDB2jWCQRN48zb5V6.CEabTPf."
emailVerified : false
sharePersonalInfo : false
accountCreated : 2025-04-22T14:31:52.334+00:00
roles : Array (1)
enabled : true
provider : "local"
```

## 7. GitHub

- GitHub Repo link  
[https://github.com/dayanakumar-IT/CosmoConnect\\_SkillSharingApp\\_PAF.git](https://github.com/dayanakumar-IT/CosmoConnect_SkillSharingApp_PAF.git)
- Image of the GitHub Commit Tree

Initial Git-Graph Commit tree Image



## Recent Git-Graph Commit tree Image

update card design with Bootstrap utilities	7 May 2025 22:19	Harishalinee Elango...	297a0077
integrate Bootstrap for basic styling	7 May 2025 22:19	Harishalinee Elango...	c9cd6cca
extract appointment form into separate component	7 May 2025 22:18	Harishalinee Elango...	7232dc32
Created a dynamic competition results page	7 May 2025 22:18	Vidursha Prabagaran	79f9aab6
implement conditional rendering for user dashboard	7 May 2025 22:18	Harishalinee Elango...	f59e8af9
correct state update logic in booking form	7 May 2025 22:18	Harishalinee Elango...	6ef365b7
create reusable button component	7 May 2025 22:17	Harishalinee Elango...	c71a8181
Added date and time picker to competition entry form	7 May 2025 22:17	Vidursha Prabagaran	7939f7b0
add interactive dropdown menu	7 May 2025 22:17	Harishalinee Elango...	50f9baaf
Fixed mobile UI issues for competition pages	7 May 2025 22:17	Vidursha Prabagaran	12eb7c77
<span style="color: green;">Merge pull request #5 from dayanakumar-IT/feature/Post-UserProfile-Comment</span>	7 May 2025 22:15	Dayana Kumar	0e58c9c6
<span style="color: orange;">Deleted Feature branch</span>	7 May 2025 22:13	Dayana Kumar	3468c4f8
<span style="color: purple;">implemented complete frontend CRUD functionality for course</span>	7 May 2025 18:24	Kaushalya Nagenth...	55a3b7dc
gave course images rounded corners	7 May 2025 18:20	Kaushalya Nagenth...	91e3814f
added divider between course info and action buttons	7 May 2025 18:20	Kaushalya Nagenth...	0e8e1c5f
styled submit and cancel buttons on course form	7 May 2025 18:20	Kaushalya Nagenth...	20090c65
minor changes	7 May 2025 18:19	Kaushalya Nagenth...	c313ac8a
updated course title font for better readability	7 May 2025 18:19	Kaushalya Nagenth...	1a0f4dfe
added consistent margin between form fields	7 May 2025 18:18	Kaushalya Nagenth...	c3ca2ae7
styled empty course list message	7 May 2025 18:18	Kaushalya Nagenth...	0e23b8ed
styled add/edit course modal with better spacing	7 May 2025 18:18	Kaushalya Nagenth...	33cac0ee
updated background color for course section	7 May 2025 18:17	Kaushalya Nagenth...	ffaf9ca6
added animation on course item hover	7 May 2025 18:17	Kaushalya Nagenth...	bcb6e225
improved checkbox and dropdown style	7 May 2025 18:16	Kaushalya Nagenth...	f4b6edc1
applied card layout to course list items	7 May 2025 18:16	Kaushalya Nagenth...	0921135d

# IT2178640 – Commit History

dayanakumar IT / CosmoConnect\_SkillSharingApp\_PAF

Code Issues Pull requests Actions Projects Security Insights Settings

Commits

feature/Post-User... All users All time

Commits on May 7, 2025

Deleted Feature branch	5468c4f		
dayanakumar IT committed 18 minutes ago			
redesign post layout for better user engagement	83abc73		
dayanakumar IT committed 5 hours ago			
Redesign post layout for better user engagement	66239da		
dayanakumar IT committed 5 hours ago			
Improved profile card layout and responsiveness	f+70e86		
dayanakumar IT committed 5 hours ago			
resolve login redirect issue after social auth	d2b1c99		
dayanakumar IT committed 5 hours ago			
Edited Space calendar alignment	37890a7		
dayanakumar IT committed 5 hours ago			
comment edit option enabled with icon enhancement	8def7fa		
dayanakumar IT committed 7 hours ago			
Delete option for comment enabled	ccc6911		
dayanakumar IT committed 7 hours ago			
Seeing previous comments for a particular post comments enabled	d27126e		
dayanakumar-IT committed 7 hours ago			
story option UI created	9826d66		
dayanakumar-IT committed 7 hours ago			
Space Calendar add in the feed	6f5186c		
dayanakumar-IT committed 7 hours ago			
Removed one of the challenge	1e1617f		
dayanakumar-IT committed 7 hours ago			
Spinning option updated	d5e4ae9		
dayanakumar-IT committed 8 hours ago			
Added new challenge question and answering option	a6eab6		
dayanakumar-IT committed 8 hours ago			
Daily Space challenge component added	2de4b97		
dayanakumar-IT committed 8 hours ago			
Edited floating background icons	7de0897		
dayanakumar-IT committed 8 hours ago			
Astronomy fact card component created	a7c5159		
dayanakumar IT committed 8 hours ago			
Upcoming space events-updated with UI enhancement	f336c8c		
dayanakumar IT committed 8 hours ago			
Followers icons and profile images adjusted	bbee5d0		
dayanakumar IT committed 8 hours ago			
Followers list alignment implemented	0575d26		
dayanakumar-IT committed 8 hours ago			
Updated the save post button	3ba1186		
dayanakumar IT committed 8 hours ago			
Like - Emoji Picker updated	a5925dc		
dayanakumar IT committed 8 hours ago			
Replaced the single like button with an emoji reaction picker	18bedc75		
dayanakumar IT committed 8 hours ago			
Posts Feed alignment adjusted	6dd56cb		
dayanakumar IT committed 8 hours ago			
Changed UI under explore navigation	31a0216		
dayanakumar-IT committed 8 hours ago			
Removed Feed Front end code for rebuilding	62864bf		
dayanakumar-IT committed 9 hours ago			
added enhancement images for UI purpose	268e1e3		
dayanakumar-IT committed 9 hours ago			
Post Feed navigation bar option updated	bed5d11		
dayanakumar-IT committed 9 hours ago			
Updated backend Delete functionality	eb96925		
dayanakumar-IT committed 9 hours ago			
Updated 3 dot and timer alignment in post UI	8cfdd7a		
dayanakumar-IT committed 10 hours ago			
Updated edit form for POSTs	ee03e2a		
dayanakumar-IT committed 10 hours ago			
Post READ enhancement for displaying uploaded images	7d45af9		
dayanakumar-IT committed 10 hours ago			
Removed Certain icons and replaced with my new icons set	0066f00		
dayanakumar-IT committed 10 hours ago			
Icon looks changed	b304563		
dayanakumar-IT committed 10 hours ago			
Added Poll option to posts	4c4c516		
dayanakumar-IT committed 10 hours ago			

Previous Next >

# IT22057488 – Commit History

Commits

feature/learning...

Commits on May 7, 2025

- switch from Bootstrap to Tailwind for styling
- add responsive navbar with dropdown menu
- resolve overlapping content in small screens
- update button design using Tailwind utilities
- adjust padding and margins for consistency
- add comments to clarify form validation logic
- simplify component structure for better readability
- update project README with setup instructions
- resolve layout shifting issue
- clean up unused Bootstrap classes
- add custom button styles using SCSS
- adjust spacing for consistent layout
- update card design with Bootstrap utilities
- integrate Bootstrap for basic styling
- extract appointment form into separate component
- implement conditional rendering for user dashboard
- correct state update logic in booking form
- create reusable button component
- add interactive dropdown menu
- Learning progress delete button
- Learning progress update button
- Refactored Header component for cleaner structure
- Added PostCard component for displaying shared posts
- Adjusted font sizes for readability on smaller screens
- Improved form layout on mobile view
- Applied consistent padding across all sections
- Updated button styles for better visibility
- The full Frontend CRUD

Commits on May 3, 2025

- learn: improved UI responsiveness
- learn: created dynamic calendar
- learn: handled form edge cases
- learn: handled form edge cases
- learn: added update/delete functionality
- learn: used useState and useEffect
- learn: explored Bootstrap styling

Previous Next >

# IT22294098 – Commit History

The screenshot shows the GitHub commit history for the repository 'CosmoConnect\_SkillSharingApp.PAF'. The commits are organized by date: May 7, 2025, and May 6, 2025. Each commit includes a message, the author (Vidursha), the commit time (e.g., 'committed 11 minutes ago' or 'yesterday'), and the commit hash.

**Commits on May 7, 2025:**

- Completed competition leaderboard styling and layout (9aa887d)
- Finished frontend layout for competition dashboard and user navigation. (7245659)
- Added navigation menu for easy access to competition pages (0f6e376)
- Implemented validation for competition entry forms (b3b4b71)
- Created a dynamic competition results page (79f9aab)
- Added date and time picker to competition entry form (793947b)
- Fixed mobile UI issues for competition pages (12eb7c7)
- Fix layout issues in competition view page (85b6451)
- Display full competition information on read page (db892dd)
- Fix alignment of icons in competition action buttons (afe2465)
- Fix alignment of icons in competition action buttons (6d5ede6)
- Update hover color for competition cards (0dbba3b)
- Change header text color on competition page (2ce31b8)
- Update form button text dynamically (8707047)
- Display competition creation date on detail view (35121fa)

**Commits on May 6, 2025:**

- Add icons to represent competition types (68a94d3)
- Update competition list to support sorting by date (db4ddaa)
- Implement dark mode support for competition pages (fc63b71)
- Integrate date validation for competition deadlines (99b755b)
- Highlight active competition in the list (fefab6a)
- Add date picker for competition start and end dates (64ha399)
- Implement search and filter on competition list (0e3f152)
- Enhance UI with icons for edit and delete actions (78998c2)
- Update styles for competition management pages (c641de8)
- Fix layout issues in competition form (0bf4f12)
- Add edit and delete buttons to competition cards (8b3fa8d)
- Update button styles in competition management UI (ebcf92e)
- Create responsive layout for competition pages (55fd873)

# IT22289384 – Commit History

The screenshot shows the GitHub commit history for the repository 'CosmoConnect\_SkillSharingApp\_PAF' under the user 'dayanakumar-IT'. The commits are filtered by the branch 'feature/learning...' and date range 'May 7, 2025'. There are 45 commits listed, all made by the user 'Kaushalyaa' 4 hours ago. The commits are organized into several logical groups:

- Initial Setup and Frontend:** Includes commits like 'initialize frontend for CosmoConnect app', 'setup project structure and install dependencies', 'implement landing page UI layout', 'add responsive user profile card component', 'create user dashboard page', 'update global styles and theme colors', 'polish chat UI for better readability', 'Changes made and corrections done', 'adjusted padding and margins across components', 'added hover effects to course button', 'made form inputs look consistent', 'fixed alignment issues on edit course page', 'Adjusted stylings for course card', 'added border and shadows to course form', 'styled empty course list message', 'improved checkbox and dropdown style', 'applied card layout to course list items', 'cleaned up margins between form sections', 'improved feedback message styling', 'aligned form labels and inputs properly', 'added border and shadows to course form', 'updated background color for course section', 'styled add/edit course modal with better spacing', 'minor changes', 'added consistent margin between form fields', 'updated course title font for better readability', 'added divider between course info and action buttons', 'gave course images rounded corners', and 'implemented complete frontend CRUD functionality for course'.
- UI Enhancements:** Focuses on improving the user interface, including styling for course lists, modals, and dropdowns.
- Form and Button Styling:** Includes commits related to styling form inputs, labels, and buttons.
- General Improvements:** Includes commits for alignment issues, padding/margins, and overall consistency.

Previous Next >

## 8. References

- Claude.ai. (n.d.). *AI-generated chords*. Retrieved from <https://www.claude.ai>.
- W3Schools. (n.d.). *HTML, CSS, JavaScript tutorials and references*. Retrieved from <https://www.w3schools.com>.
- Tailwind CSS. (n.d.). *Tailwind CSS Documentation*. Retrieved from <https://tailwindcss.com/docs>.
- HTML5 UP. (n.d.). *Free HTML5 website templates*. Retrieved from <https://html5up.net>.