

illumination

Sprint 2

By Team 1 Tech Titans



Agenda

Team Member Roles and Responsibilities

Improvements made from Professor Feedback

Project Description

Team Working Agreement

Personas (at least 3)

MVP (Minimum Viable Product)

Technologies

Algorithms

Diagrams

Product Backlog

Sprint 2 Backlog

Metrics

Retrospective

Sprint 3 Plan

Project Demo - Sprint 2

GitHub Link

Live Application Demo



Team Members



**Anuhyaa
Marapalli**

Scrum Master /
Developer



**Snehalatha
Boothpur**

Designer / Developer



**Rithin Guptha
Bajuri**

Developer



Team Members



**Harshitha
Rangaraju**

Team Leader and
Developer



**Srinivas Reddy
Bapathu**

Developer and Tester



Improvements

- Creating Proper Burndown Chart
- Change in the context diagram

Project Description



Project Name:	Illumination
Team:	Tech Titans
Project Description:	<p>Illumination is an AI-driven mobile learning application designed to deliver personalized learning through real-time adjustment of article contents based on students' performances and preference.</p> <p>For students who wants more structured learning experience the Illumination</p> <p>is a app with it's advanced ML techniques, that adjusts the difficulty and provides personalized content recommendations, unlike traditional learning apps or already pre defined content application</p> <p>our application uses AI to continuously monitor student learning and it dynamically updates the learning path.</p>
Benefit Outcomes:	<ul style="list-style-type: none">• This will improve the understanding of the student in weak areas.• This continuous tracking, will also get personalized feedback, which will let students know where they lag.• This will improve students retention through spaced repetition.• This is also way more efficient in learning new concepts.
Github Link:	https://github.com/htmw/2024F-Tech-Titans/wiki

Team Agreement



Team Agreement

Team Tech Titans

- We, as members of the team, are committed to attending all scheduled meetings on time. When this is not possible, it should be communicated to the group in advance so that proper readjustment may occur.
- If a member is unable to attend a meeting, they will inform the team beforehand and stay updated on any decisions made during their absence. In situations where rescheduling isn't feasible, the absent member agrees to follow the majority decision.
- Every member is encouraged to seek help from the other members when they have any doubts or are facing issues, instead of waiting till the last minute.
- Team members can freely express their opinions and suggestions during meetings or discussions. In case, if they didn't give feedback, their decision is taken as Yes.
- Members are encouraged to be active in the discussions or meetings and even pay attention during key discussions. It is expected that everyone will be active and participate.
- Fair distribution of tasks in the group; each member will finish his/her part of the task by the agreed deadline, so that the work on the project proceeds consistently.
- It is expected that teammates respect the time and commitment of one another. Members should be punctual with responses in the group chat and adopt a professional attitude. Every member will try to put forth their best effort.



Ankit, 17 year old

Ankit is a high schooler who is preparing for IIT JEE exams where he is good at Mathematics and Chemistry but he struggles in Physics, even though he attends extra classes, but he never excels it. So, he feels like he needs personalized study material to improve his weak areas.

Goals:

- Have good grasp on difficult physics topics.
- Have access to personalized materials on complex topics or topics where he is weak at.
- Able to balance school, extra classes and self study.

Challenges

- There is a lot of content online or in books but unable to find proper content for his weakness topics.
- Searching for proper material is time taking.
- Unable to identify the weak topics.

Rajat, 28 year old

Rajat is CA (Chartered Accountant) aspirant from Mumbai, but he is also working full time in financial firm, which causes him have less time to study. He finds financial reporting and taxation difficult and even struggling to focus because of tight schedule.

Goals:

- Improve the understanding of the complex topics.
- Able to access the targetted material on the topics which he finds difficult.
- Able to balance both work and preparation.

Challenges:

- Its difficult to manage both work life and student life which leading him not to study well.
- Time consuming on finding the concepts he is weak at.
- Struggling to study all because of broad topics in CA.





Kavya, 21 year old

Kavya is engineering student, as she is near her job trials and she is interested in Machine Learning and GenAI, she want to study them and learn them, but she have no idea on machine learning. When she searching of google for roadmap, all those roadmap doesn't suit her well which causes her to distract.

Goals

- Master all Machine learning and deep learning concepts.
- Able to read fine quality materials which simplify the complex concepts into simple once.

Challenges

- There are alot of materials on it which causes to read alot of materials of same topics to understand them.
- Facing difficulties in retaining the concepts.
- Unable to keep up with the latest trends on ML.

MVP

- **Subject Selection:** Choose subjects to access articles.
- **Article Reading:** Browse and read subject-specific articles.
- **Mark Articles for Later:** Save articles for future review.
- **Search Functionality:** Search articles within a subject.
- **Personalized Recommendations:** Get suggestions based on performance.

Technologies

01.

Backend

fastAPI
Node JS
Mongodb
AWS
Pytorch

02.

Frontend

React Native

03.

Tools

Github
Postman
Google Colab
VSCode

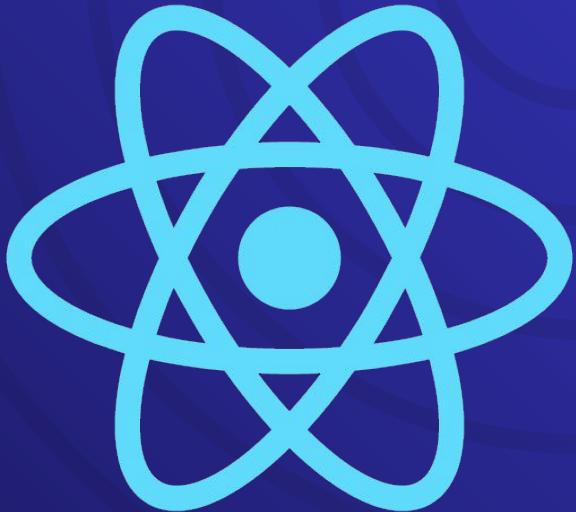
1. Backend

Here FastAPI is primarily used for machine learning backend tasks where as Node JS is primarily deals with Client Side code. MongoDB for Storing the data. AWS for hosting and even for storing the objective Data. Pytorch for Machine learning



2. Frontend

React Native is used for building cross platform mobile application.



3. Tools

Github is where code and even the documents are shared. Postman is for API Testing. Google Colab for training the Machine Learning Model. VSCode is used for writing the code





Algorithms

Reinforcement Learning

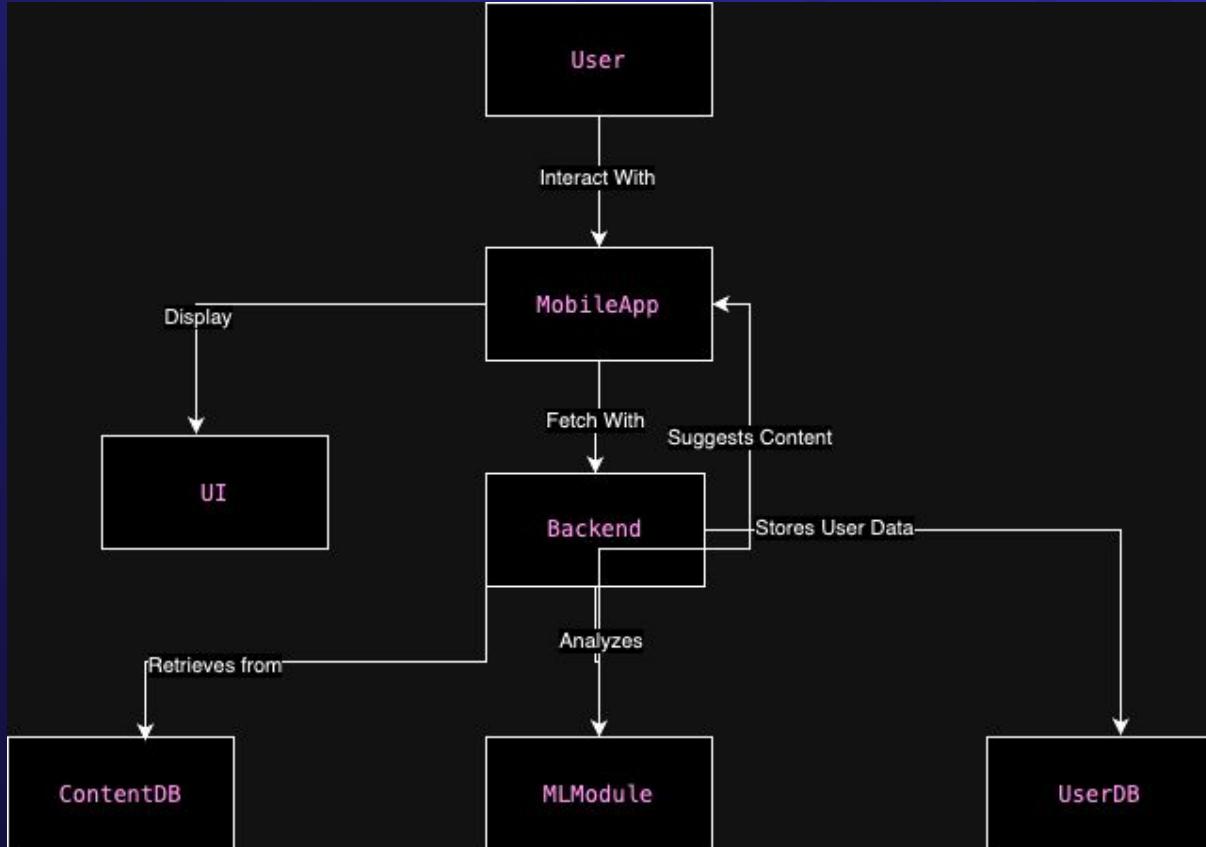
Reinforcement Learning (RL) is a machine learning paradigm where an agent learns by interacting with an environment and receiving rewards for actions. In adaptive learning, RL is used to dynamically adjust content difficulty based on student responses, maximizing engagement and learning efficiency over time.

Word2Vec

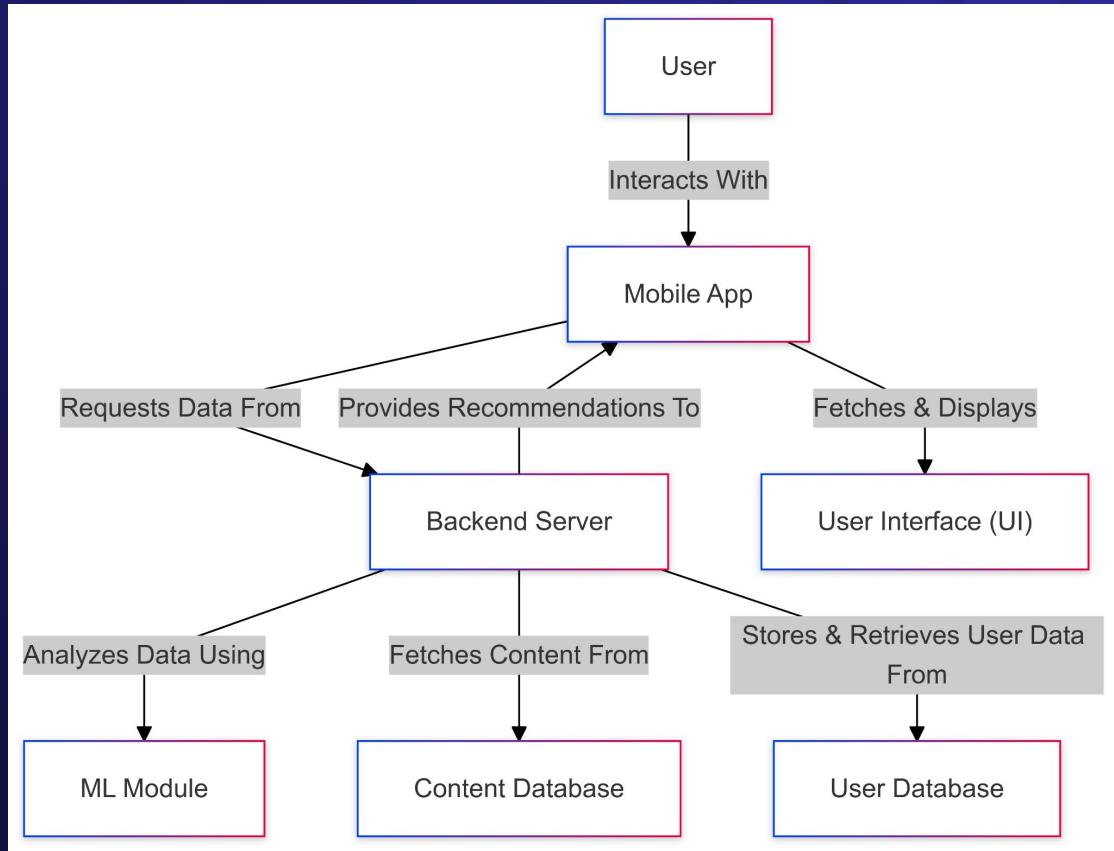
Word2Vec is a deep learning-based model that transforms words into continuous vector representations, capturing semantic relationships between words. In content-based filtering, it is used to convert textual content (article text) into vectors, allowing us to measure similarity between content items. Similar vectors imply similar content, enabling personalized recommendations.



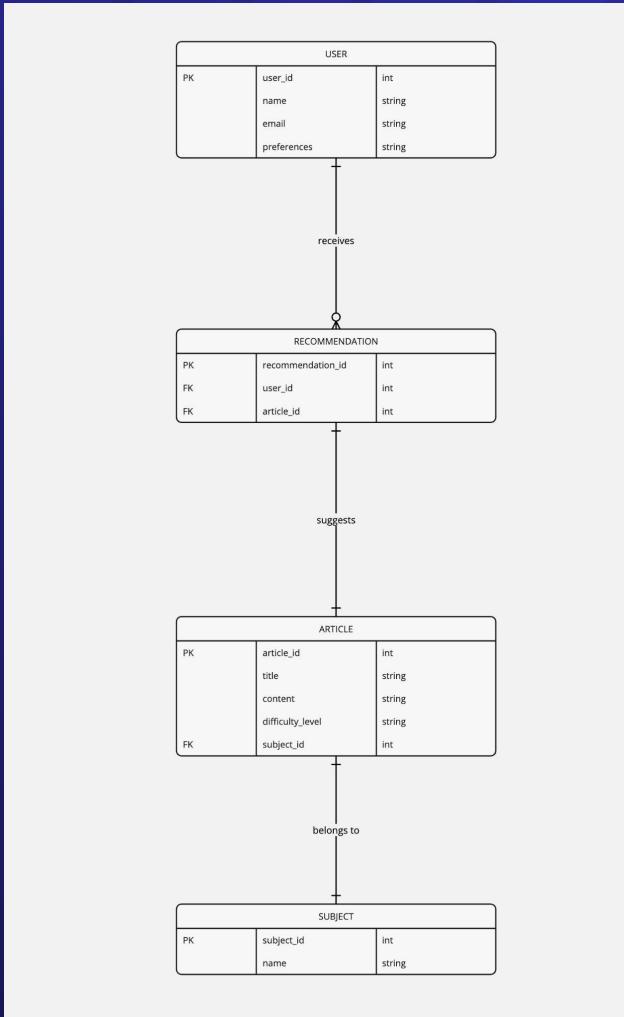
Architecture Diagram



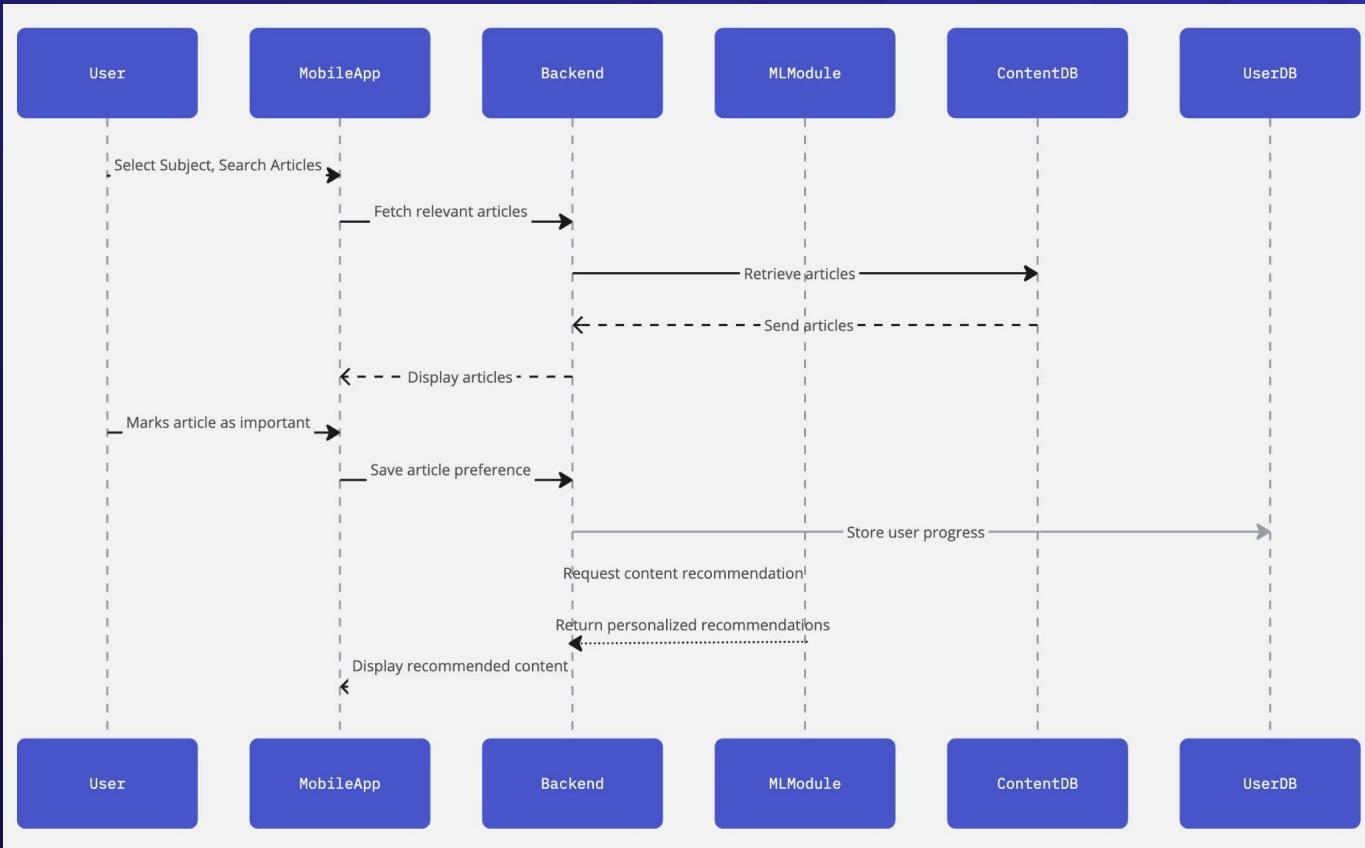
Context Diagram



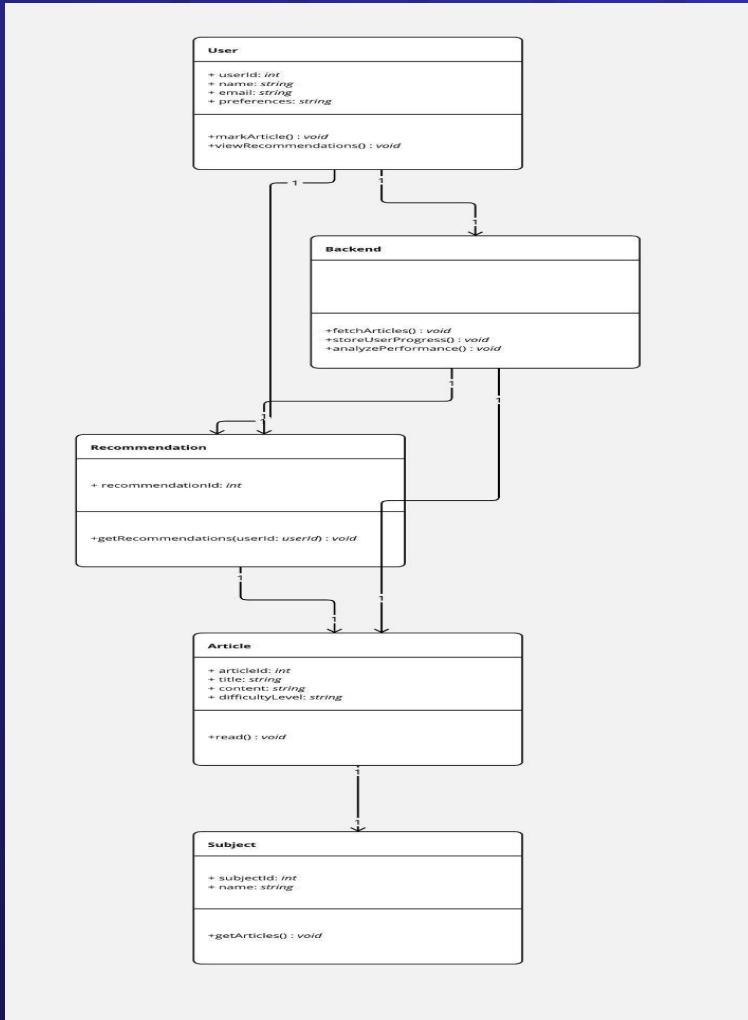
ER Diagram



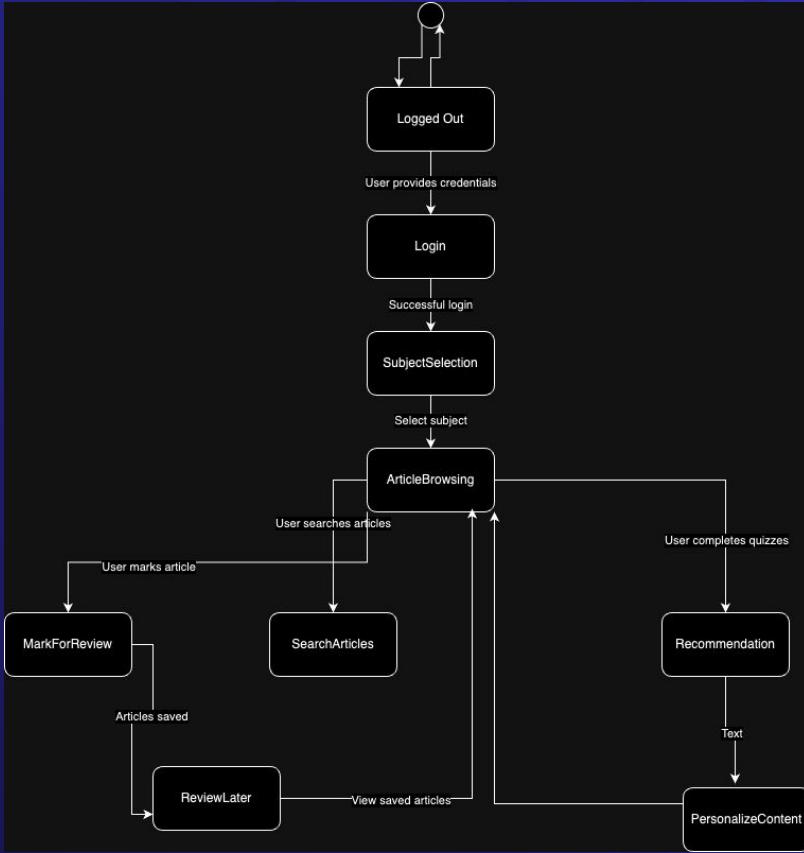
Sequence Diagram



Class Diagram



State Diagram





Sprint 1 Recap

- **Subject Selection and Article Reading Features:** Enabled selecting subjects and reading relevant articles smoothly.
- **Mark, Search, and Navigate Articles:** Allowed marking, searching, and seamless navigation between articles.
- **Basic UI and Content Management Setup:** Developed clean interface and efficient content storage system.



Product Backlog

#	Sprint	Feature	User Story/Technical Story	Acceptance Criteria
1	Sprint 1	Subject Selection	User Story 1: As a student, I want to select a subject so I can access relevant learning content.	Users can select from multiple subjects (e.g., Math, Science, History). Selected subjects load relevant articles.
2	Sprint 1	Article Reading	User Story 2: As a student, I want to browse and read articles related to the subject I selected.	Articles display in a clean, mobile-friendly interface. Users can scroll and navigate within articles seamlessly.
3	Sprint 1	Mark Articles for Later	User Story 3: As a student, I want to mark articles as "difficult" or "important" for later review.	Users can mark articles as "difficult" or "important". Marked articles appear in a "Review Later" section.
4	Sprint 1	Article Search	User Story 4: As a student, I want to search for specific articles within a subject.	Users can search for articles by keywords within a subject. Search results are displayed based on relevance.
5	Sprint 1	Article Navigation 	User Story 5: As a student, I want to navigate between articles within the same subject.	Users can use "next" and "previous" buttons or a navigation bar to move between articles. Ensure smooth transitions.

Product Backlog

#	Sprint	Feature	User Story/Technical Story	Acceptance Criteria
6	Sprint 1	Basic UI for Articles	Technical Story 1: Set up basic UI/UX for browsing and reading articles.	Clean, intuitive design for subject selection and article reading. Ensure mobile responsiveness.
7	Sprint 1	Content Management System	Technical Story 2: Implement content management system for loading articles by subject.	Articles are stored and categorized by subject. Efficient retrieval of content from a backend system.
8	Sprint 1	Mark and Save Articles	Technical Story 3: Add functionality for marking and saving articles for later review.	Users can save marked articles to a "Review Later" list. Data is persisted across sessions.
9	Sprint 1	Search Functionality	Technical Story 4: Implement article search functionality.	Enable a search feature within each subject. Search results are displayed quickly and accurately.
10	Sprint 2	Personalized Recommendations	User Story 6: As a student, I want personalized content recommendations based on my performance so I can improve weak areas.	The system provides content recommendations based on quiz performance and article difficulty. Users are directed to articles covering weak areas.

Product Backlog

#	Sprint	Feature	User Story/Technical Story	Acceptance Criteria
11	Sprint 2	Spaced Repetition	User Story 7: As a student, I want the app to implement spaced repetition to reinforce key concepts over time.	The app identifies and schedules spaced repetition for topics users struggle with. Users are prompted to revisit articles or take quizzes at optimal intervals.
12	Sprint 2	Performance Dashboard	User Story 8: As a student, I want to track my progress in a performance dashboard.	The dashboard shows quiz scores, articles read, and time spent on each subject. Users can see areas of strength and weakness visually.
13	Sprint 2	Targeted Feedback on Quizzes	User Story 9: As a student, I want to receive targeted feedback after quizzes so I can understand my mistakes.	Detailed feedback is provided for each quiz question. Feedback explains why certain answers are correct/incorrect.
14	Sprint 2	Dynamic Content Adjustment	User Story 10: As a student, I want the app to dynamically adjust content difficulty based on my performance.	The difficulty of content increases if the user performs well. If a student struggles, simpler articles or review content is recommended.
15	Sprint 2	Basic Machine Learning for Recommendations	Technical Story 5: Implement a basic machine learning model for personalized content recommendations.	The model analyzes quiz performance and article difficulty to recommend personalized learning paths. Adjustments are made based on user progress.

Product Backlog

#	Sprint	Feature	User Story/Technical Story	Acceptance Criteria
16	Sprint 2	Spaced Repetition Algorithm	Technical Story 6: Integrate spaced repetition algorithm to reinforce key concepts.	Use spaced repetition techniques to schedule when users should revisit difficult topics. Ensure the system prompts users at the right time for maximum retention.
17	Sprint 2	Performance Dashboard Development	Technical Story 7: Build a performance dashboard to visualize progress.	The dashboard shows quiz results, time spent, and subject mastery. Visual charts display performance trends.
18	Sprint 2	Quiz Feedback Mechanism	Technical Story 8: Provide detailed quiz feedback for user learning.	Implement feedback mechanisms to explain quiz results. Feedback is both constructive and designed to reinforce learning.
19	Sprint 3	User Accounts & Syncing	User Story 11: As a student, I want to create an account and log in so I can save my progress and preferences across devices.	Users can create accounts with email/password. User progress, preferences, and recommendations are saved and synced across devices.
20	Sprint 3	Adaptive Study Paths	  User Story 12: As a student, I want my study path to adapt based on my long-term learning trends and preferences.	The app uses ML to analyze long-term trends and dynamically adjust learning paths. Preferences are stored, and content is adapted based on user learning styles.

Product Backlog

#	Sprint	Feature	User Story/Technical Story	Acceptance Criteria
21	Sprint 3	Reminders for Spaced Repetition & Recommendations	User Story 13: As a student, I want to receive reminders for spaced repetition and personalized study recommendations.	Notifications prompt users for spaced repetition and recommended articles. Reminders are personalized based on past performance and study schedules.
22	Sprint 3	Learning History	User Story 14: As a student, I want to view my learning history so I can review previously studied articles and performance.	Users can access their study history, including completed articles and quizzes. History includes quiz scores, article read dates, and time spent.
23	Sprint 3	Adaptive Feedback Based on Trends	User Story 15: As a student, I want to receive adaptive feedback based on my performance trends over time so I can improve continuously.	The app provides adaptive feedback based on overall performance and trends. Feedback is more tailored as the system learns from the user's progress and actions.
24	Sprint 3	User Authentication	Technical Story 9: Implement user authentication and profile management.	Secure user authentication system with account creation, login, and password recovery. Ensure user progress and data are stored across sessions.
25	Sprint 3	Refined ML for Long-Term Learning	Technical Story 10: Refine the machine learning model for adaptive feedback based on long-term learning trends.	The ML model adapts based on user learning patterns and feedback. Ensure the model evolves based on long-term trends and personalized learning data.

Product Backlog

#	Sprint	Feature	User Story/Technical Story	Acceptance Criteria
26	Sprint 3	Reminders & Notifications	Technical Story 11: Implement reminders and notifications for spaced repetition and content recommendations.	Notifications are sent based on spaced repetition schedules and content recommendations. Ensure reminders are relevant and personalized to user learning paths.
27	Sprint 3	Final UI/UX Improvements	Technical Story 12: Finalize UI/UX improvements for a polished, engaging user experience.	Refine the user interface for a smooth, intuitive experience. Ensure the app is fully responsive and accessible on different devices.



Sprint 2

#	Sprint	Story/Task	Story Points	Acceptance Criteria
1	Sprint 2	User Story 6: As a student, I want personalized content recommendations based on my performance so I can improve weak areas.	8 Points	<ul style="list-style-type: none">- The system provides content recommendations based on quiz performance and article difficulty.- Users are directed to articles covering weak areas.
2	Sprint 2	User Story 7: As a student, I want the app to implement spaced repetition to reinforce key concepts over time.	8 Points	<ul style="list-style-type: none">- The app identifies and schedules spaced repetition for topics users struggle with.- Users are prompted to revisit articles or take quizzes at optimal intervals.
3	Sprint 2	User Story 8: As a student, I want to track my progress in a performance dashboard.	5 Points	<ul style="list-style-type: none">- The dashboard shows quiz scores, articles read, and time spent on each subject.- Users can see areas of strength and weakness visually.
4	Sprint 2	User Story 9: As a student, I want to receive targeted feedback after quizzes so I can understand my mistakes.	5 Points	<ul style="list-style-type: none">- Detailed feedback is provided for each quiz question.- Feedback explains why certain answers are correct/incorrect.
5	Sprint 2	User Story 10: As a student, I want the app to dynamically adjust content difficulty based on my performance.	8 Points	<ul style="list-style-type: none">- The difficulty of content increases if the user performs well.- If a student struggles, simpler articles or review content is recommended.

Sprint 2

#	Sprint	Story/Task	Story Points	Acceptance Criteria
6	Sprint 2	Technical Story 5: Implement a basic machine learning model for personalized content recommendations.	8 Points	<ul style="list-style-type: none">- The ML model analyzes quiz performance and article difficulty to recommend personalized learning paths.- Adjustments are made based on user progress.
7	Sprint 2	Technical Story 6: Integrate spaced repetition algorithm to reinforce key concepts.	8 Points	<ul style="list-style-type: none">- Use spaced repetition techniques to schedule when users should revisit difficult topics.- Ensure the system prompts users at the right time for maximum retention.
8	Sprint 2	Technical Story 7: Build a performance dashboard to visualize progress.	5 Points	<ul style="list-style-type: none">- The dashboard shows quiz results, time spent, and subject mastery.- Visual charts display performance trends.
9	Sprint 2	Technical Story 8: Provide detailed quiz feedback for user learning.	5 Points	<ul style="list-style-type: none">- Implement feedback mechanisms to explain quiz results.- Feedback is both constructive and designed to reinforce learning.



Test Cases

Story/Task	Test Case ID	Test Case Description	Preconditions	Steps	Expected Result	Status
User Story 6	TC6-1	Verify that content recommendations are based on quiz performance and article difficulty.	User has completed at least one quiz.	<ol style="list-style-type: none">Complete a quiz with some questions answered incorrectly.Navigate to the recommendations section.	Recommended articles focus on weak areas based on quiz performance and difficulty matches the knowledge gap.	Pass
	TC6-2	Verify that users are directed to articles covering weak areas.	User has identified weak areas through quizzes.	<ol style="list-style-type: none">Complete a quiz and check the weak areas identified.View the recommended articles for weak topics.	Recommendations specifically address topics in which the user scored poorly in quizzes.	Pass
User Story 7	TC7-1	Verify spaced repetition scheduling for topics where the user struggles.	User has a history of quiz attempts.	<ol style="list-style-type: none">Complete a quiz with some incorrect answers.Track spaced repetition scheduling for those topics.	The app schedules reminders for topics with low scores, prompting users to revisit the content.	Pass
	TC7-2	Verify that users are prompted to revisit content at optimal intervals.	User has multiple quizzes with varying results.	<ol style="list-style-type: none">Track the system prompts over a period.Check the timing of prompts to revisit specific topics.	System prompts align with spaced repetition intervals to maximize retention.	Pass



Test Cases

Story/Task	Test Case ID	Test Case Description	Preconditions	Steps	Expected Result	Status
User Story 8	TC8-1	Verify display of quiz scores, articles read, and time spent on each subject in the dashboard.	User has quiz results and article reading history.	1. Access the performance dashboard. 2. Review quiz scores, articles read, and time spent.	All information is displayed correctly and reflects the user's history accurately.	Pass
	TC8-2	Verify that areas of strength and weakness are displayed visually.	User has completed multiple quizzes.	1. Open the dashboard. 2. Analyze visual representations of strengths and weaknesses.	Visuals clearly depict areas where the user is strong or needs improvement.	Pass
User Story 9	TC9-1	Verify that detailed feedback is provided for each quiz question.	User completes a quiz.	1. Complete a quiz. 2. Review the feedback provided for each question.	Each question has detailed feedback, explaining correct/incorrect answers.	Fail
	TC9-2	Verify that feedback explains why answers are correct or incorrect.	User completes a quiz.	1. Complete a quiz and access feedback. 2. Check explanations for correctness of answers.	Feedback is clear and instructional, explaining the rationale behind answers.	Fail



Test Cases

Story/Task	Test Case ID	Test Case Description	Preconditions	Steps	Expected Result	Status
User Story 10	TC10-1	Verify that content difficulty increases with good performance.	User consistently scores well in quizzes.	<ol style="list-style-type: none">Complete multiple quizzes with high scores.Check the difficulty level of subsequent content recommendations.	Recommended content has a higher difficulty level after strong performance in quizzes.	Fail
	TC10-2	Verify that simpler articles are recommended if the user struggles.	User performs poorly on multiple quizzes.	<ol style="list-style-type: none">Complete quizzes with low scores.View the recommended content following poor performance.	Recommended content is simplified, focusing on foundational topics.	Fail
Technical Story 5	TC5-1	Verify that the ML model generates personalized recommendations based on quiz performance.	ML model is integrated into the system.	<ol style="list-style-type: none">Complete a quiz with varied results.Access recommended content.Check if recommendations align with weak and strong areas.	Recommendations are tailored based on quiz performance, focusing on weak areas identified by the ML model.	Pass
	TC5-2	Verify that the ML model adjusts recommendations as user performance changes.	ML model is active and collecting data.	<ol style="list-style-type: none">Complete multiple quizzes with different scores.Monitor any changes in the type and difficulty of recommended content.	Recommended content adjusts in difficulty and relevance as user performance improves or declines.	Pass

Test Cases

Story/Task	Test Case ID	Test Case Description	Preconditions	Steps	Expected Result	Status
Technical Story 6	TC6-1	Verify that spaced repetition prompts are generated based on user quiz performance.	Spaced repetition algorithm is configured.	<ol style="list-style-type: none">Complete quizzes with some incorrect answers.Observe when the system prompts a review of these topics.Verify prompt timing.	Users receive reminders for challenging topics at appropriate intervals to reinforce memory and concept retention.	Pass
	TC6-2	Verify that spaced repetition scheduling considers user performance trends.	Spaced repetition algorithm is active.	<ol style="list-style-type: none">Complete multiple quizzes on the same topic with varied performance.Check if the system dynamically updates the review schedule based on recent performance.	Review schedule adjusts based on user's latest performance, prioritizing difficult topics when needed.	Pass
Technical Story 7	TC7-1	Verify that quiz results, time spent, and subject mastery are displayed on the dashboard.	User has completed quizzes and read articles.	<ol style="list-style-type: none">Access the performance dashboard.Check for accurate display of quiz scores, time spent, and subject mastery metrics.	Dashboard displays comprehensive data that accurately reflects the user's progress and engagement.	Pass
	TC7-2	Verify that visual charts display trends in user performance over time.	+ User has accumulated quiz and article data.	<ol style="list-style-type: none">Open the performance dashboard.Review visual charts for clarity and accuracy in representing performance trends.	Visualizations provide an accurate, easily interpretable view of user performance trends.	Pass

Test Cases

Story/Task	Test Case ID	Test Case Description	Preconditions	Steps	Expected Result	Status
Technical Story 8	TC8-1	Verify that each quiz question includes detailed feedback on correct and incorrect answers.	User has completed a quiz.	<ol style="list-style-type: none">1. Complete a quiz.2. Review feedback for each question.3. Confirm that explanations are provided for all answers.	Feedback is detailed and provides constructive explanations, enhancing the user's understanding.	Fail
	TC8-2	Verify that feedback helps users understand their mistakes and improve.	User has completed multiple quizzes.	<ol style="list-style-type: none">1. Access feedback after completing several quizzes.2. Observe if the feedback explains misconceptions or knowledge gaps.	Feedback is instructional and encourages better comprehension and retention of correct concepts.	Fail



Sprint 2 Completed Stories

Story/Task	Story Points
User Story 6: Personalized Content Recommendations	8 Points
User Story 7: Spaced Repetition Implementation	8 Points
User Story 8: Performance Dashboard	5 Points
Technical Story 5: Machine Learning Model for Recommendations	8 Points
Technical Story 6: Spaced Repetition Algorithm	8 Points
Technical Story 7: Performance Dashboard Visualization	5 Points

Sprint 2 Not Completed Stories

Story/Task	Story Points
User Story 9: Targeted Feedback After Quizzes	5 Points
User Story 10: Dynamic Content Difficulty Adjustment	8 Points
Technical Story 8: Detailed Quiz Feedback	5 Points

Team Velocity - This Sprint

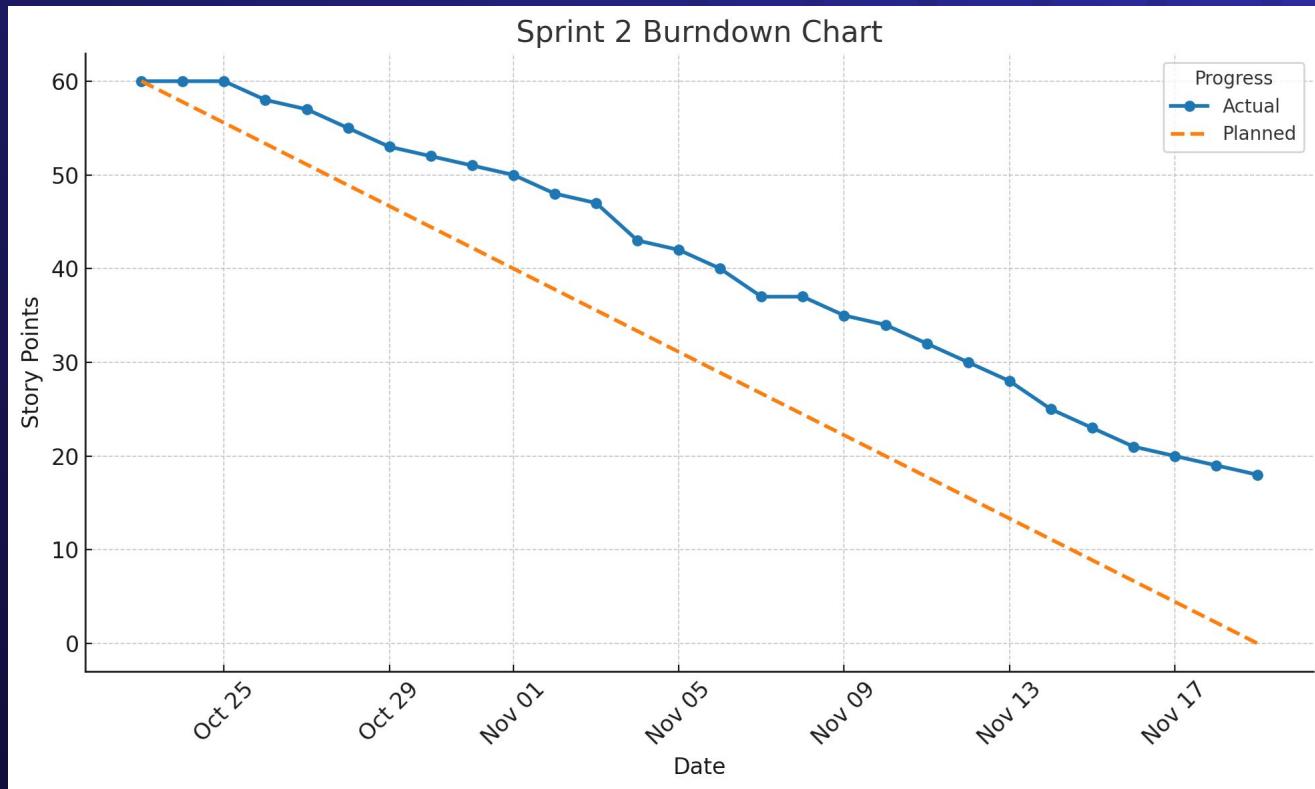
Total Story Points Completed: 42 Points

Team's Historical Velocity (Average)

Historical Velocity (Average): Calculated by averaging the total points completed across previous sprints (excluding Sprint 0).

- **Sprint 1:** 42 points
- **Sprint 2:** 42 points
- **Average Velocity:** 42.0 story points

Burndown Chart - Sprint 2



Completed/Committed Ratio

- **Committed:** 60 story points
- **Completed:** 42 story points
- **Ratio:** $(42 / 60) * 100 = 70\%$



Completed/Committed Ratio - Average and Historical

- **Sprint 1:** 100% completion (42 out of 42 points)
 - **Sprint 2:** 70% completion (42 out of 60 points)
 - **Average Completed/Committed Ratio:** $(100 + 70) / 2 = 85\%$
- 



Retrospective

Sprint 2

What went well +		What can be improved +		Action Items +	
Good use of tools + 3	Quality code + 2	More efficient meetings + 2	we had less meetings than required + 1	updating sprint task daily. + 0	start of sprint 3 early + 0
Sprint planning and successfully recorded + 1	we have chosen right technologies + 5	Team Collaboration + 2	understanding of requirements and implementation + 2	Need to spend more time on researching to meet expectations + 0	planning the meeting schedule before. + 0
continuous improvement + 3	we had improved in sticking with work agreement + 3	Better understanding of connectivity between frontend and backend + 2	need to focus on main tasks + 2		
team has encountered the challenges + 2	Problem-Solving + 3	Better workload Distribution + 1	checking and updating each progress regularly. + 3		
team helped each other resolving conflicts. + 16		Better estimation and planning + 3	Time management + 0		

Sprint 3



#	Sprint	Story/Task	Story Points	Acceptance Criteria
1	Sprint 3	User Story 11: User Accounts & Syncing	8 Points	Users can create accounts with email/password. User progress, preferences, and recommendations are saved and synced across devices.
2	Sprint 3	User Story 12: Adaptive Study Paths	13 Points	The app uses ML to analyze long-term trends and dynamically adjust learning paths. Preferences are stored, and content is adapted based on user learning styles.
3	Sprint 3	User Story 13: Reminders for Spaced Repetition & Recommendations	5 Points	Notifications prompt users for spaced repetition and recommended articles. Reminders are personalized based on past performance and study schedules.
4	Sprint 3	User Story 14: Learning History	8 Points	Users can access their study history, including completed articles and quizzes. History includes quiz scores, article read dates, and time spent.
5	Sprint 3	User Story 15: Adaptive Feedback Based on Trends	8 Points	The app provides adaptive feedback based on overall performance and trends. Feedback is more tailored as the system learns from the user's progress and actions.

Sprint 3

#	Sprint	Story/Task	Story Points	Acceptance Criteria
6	Sprint 3	Technical Story 9: User Authentication	3 Points	Secure user authentication system with account creation, login, and password recovery. Ensure user progress and data are stored across sessions.
7	Sprint 3	Technical Story 10: Refined ML for Long-Term Learning	8 Points	The ML model adapts based on user learning patterns and feedback. Ensure the model evolves based on long-term trends and personalized learning data.
8	Sprint 3	Technical Story 11: Reminders & Notifications	5 Points	Notifications are sent based on spaced repetition schedules and content recommendations. Ensure reminders are relevant and personalized to user learning paths.
9	Sprint 3	Technical Story 12: Final UI/UX Improvements	3 Points	Refine the user interface for a smooth, intuitive experience. Ensure the app is fully responsive and accessible on different devices.
10	Sprint 2 (Carryover)	User Story 9: Targeted Feedback on Quizzes	5 Points	Detailed feedback is provided for each quiz question. Feedback explains why certain answers are correct/incorrect.

Sprint 3

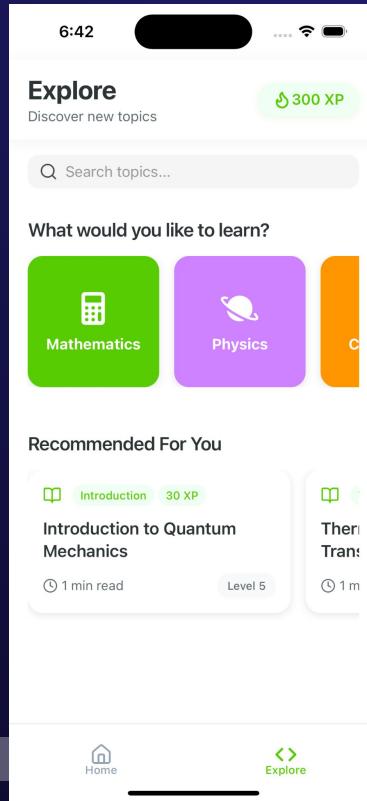


#	Sprint	Story/Task	Story Points	Acceptance Criteria
11	Sprint 2 (Carryover)	User Story 10: Dynamic Content Adjustment	8 Points	The difficulty of content increases if the user performs well. If a student struggles, simpler articles or review content is recommended.
12	Sprint 2 (Carryover)	Technical Story 8: Quiz Feedback Mechanism	5 Points	Implement feedback mechanisms to explain quiz results. Feedback is both constructive and designed to reinforce learning.

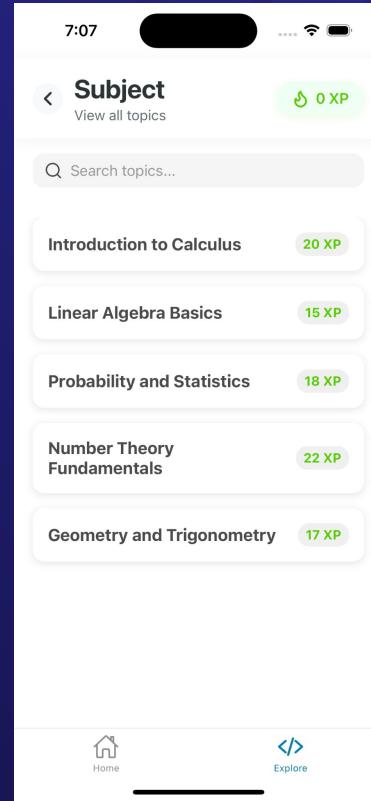


App Screenshots

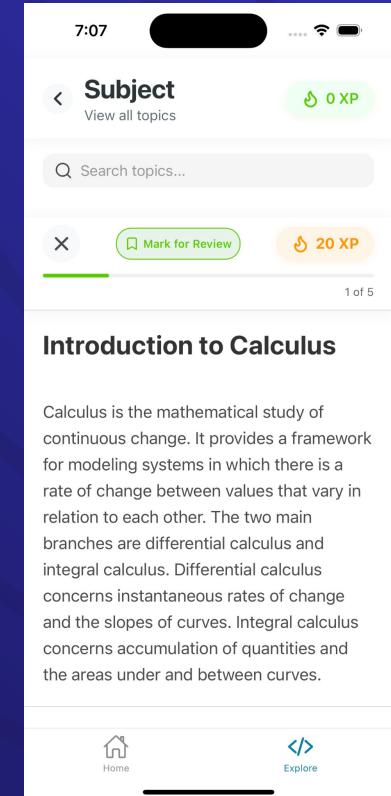
Explore Screen



List of Lessons

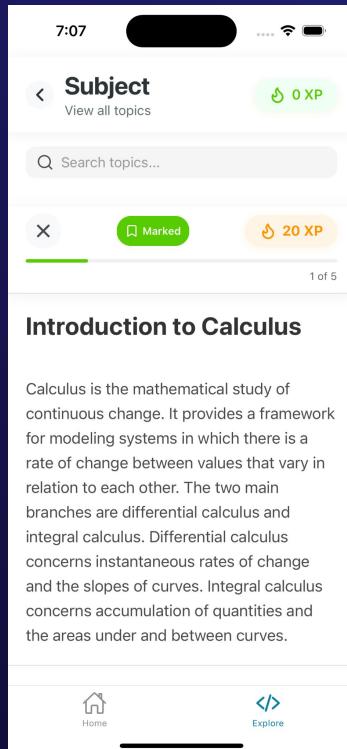


Article screen

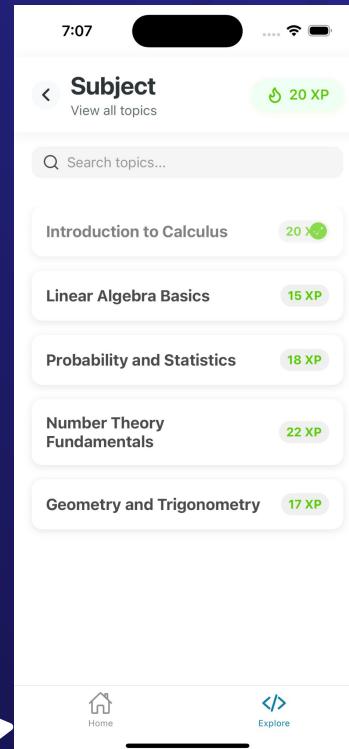


App Screenshots

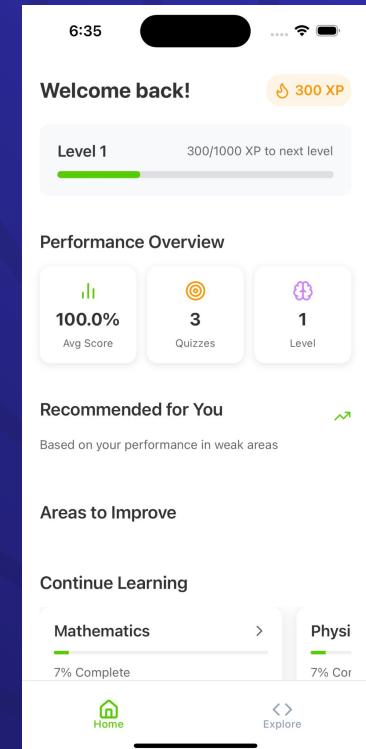
When article is marketed to read later



When article is completed, adding of rewards

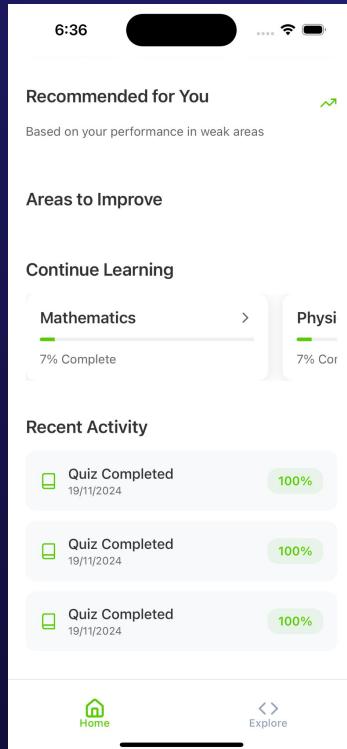


HomeScreen with dashboard

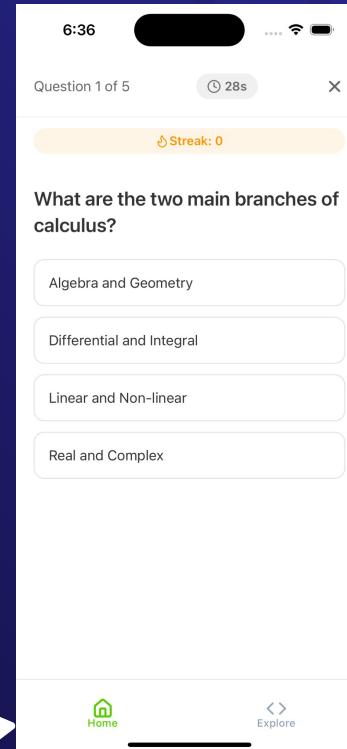


App Screenshots

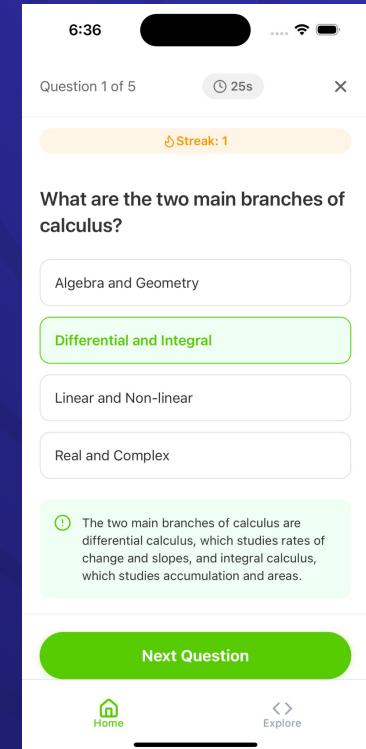
Recent Activities



Quiz

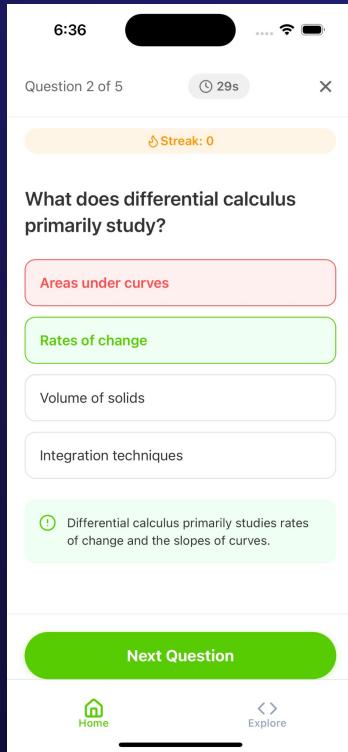


When correct answer



App Screenshots

When answered wrong



API

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', 'API Network', a search bar 'Search Postman', and various account and settings icons. A prominent 'Upgrade' button is visible.

The main workspace is titled 'Team Workspace'. On the left, there's a sidebar with 'Collections' (empty), 'Environments' (empty), and 'History' (empty). The central area shows an 'Overview' tab for a collection named 'GET http://localhost:3000/api/*'. Below it, a request card is displayed for 'http://localhost:3000/api/health' using a 'GET' method.

The request card includes tabs for 'Params', 'Authorization', 'Headers (7)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. Under 'Headers', there are seven entries. The 'Body' tab is selected, showing a table for 'Query Params' with columns 'Key', 'Value', 'Description', and 'Bulk Edit'. There are no entries in this table.

At the bottom of the request card, status information is shown: 'Status: 200 OK', 'Time: 37 ms', 'Size: 281 B', and a 'Save as example' button. Below this, the 'Body' tab displays the response body in 'Pretty' format:

```
1 [ {  
2   "status": "ok"  
3 } ]
```

At the very bottom of the screen, there are several small status indicators and links: 'Online', 'Find and replace', 'Console', 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', and a few other icons.

API

The screenshot shows the Postman application interface. At the top, there's a navigation bar with Home, Workspaces, API Network, a search bar for 'Postman', and various user account options like 'Invite', 'Upgrade', and 'Logout'. Below the header is a left sidebar titled 'Team Workspace' containing sections for Collections, Environments, and History. The main workspace is titled 'http://localhost:3000/api/subjects' and shows a 'GET' request. The 'Params' tab is selected, showing a single 'Key' entry. The 'Headers' tab lists 7 items. The 'Body' tab is collapsed. The 'Tests' and 'Settings' tabs are also present. On the right side, there are buttons for 'Save', 'Send', and 'Cookies'. Below the request details, the response section is visible, showing a status of 200 OK, a response time of 22 ms, and a size of 464 B. The response body is displayed in JSON format, showing a list of subjects:

```
[{"id": "1", "name": "Mathematics", "icon": "calculator", "color": "#58CC02"}, {"id": "2", "name": "Physics", "icon": "planet", "color": "#CE82FF"}, {"id": "3", "name": "Chemistry", "icon": "flask", "color": "#F0A0A0"}, {"id": "4", "name": "Biology", "icon": "microscope", "color": "#A0D0A0"}, {"id": "5", "name": "Geography", "icon": "globe", "color": "#80C0A0"}, {"id": "6", "name": "History", "icon": "book", "color": "#C0A0E0"}, {"id": "7", "name": "Art", "icon": "paintbrush", "color": "#A0A0F0"}]
```

At the bottom of the interface, there are various utility buttons: Online, Find and replace, Console, Postbot, Runner, Start Proxy, Cookies, Vault, Trash, and Help.

API

The screenshot shows the Postman application interface. At the top, there's a navigation bar with Home, Workspaces, API Network, a search bar, and various global settings like Invite, Upgrade, and environment selection. The main workspace is titled "Team Workspace". On the left sidebar, there are sections for Collections, Environments, and History. The central area displays a request configuration for a GET method to the URL `http://localhost:3000/api/subjects/1/articles`. The "Params" tab is selected, showing a single query parameter named "Key". Below the request, the response section is visible, showing a status of 200 OK with a response body containing two articles.

GET http://localhost:3000/api/subjects/1/articles

Params

Key	Value	Description
Key	Value	Description

Body

```
[{"id": "m1", "subjectId": "1", "title": "Introduction to Calculus", "content": "Calculus is the mathematical study of continuous change...", "xp": 20}, {"id": "m2", "subjectId": "1", "title": "Linear Algebra Basics", "content": "Linear algebra is the branch of mathematics concerning linear equations...", "xp": 15}]
```

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 12 ms Size: 567 B Save as example

Online Find and replace Console

Postbot Runner Start Proxy Cookies Vault Trash

API

The screenshot shows the Postman application interface. On the left, there's a sidebar titled "Team Workspace" with sections for Collections, Environments, and History. The main area is titled "Overview" and shows a "POST" request to "http://localhost:3000/api/review-marks". The "Body" tab is selected, displaying a JSON response:

```
1 {
2   "userId": "user1",
3   "markedArticles": [
4     "m1"
5   ]
6 }
```

At the bottom, there are navigation links for "Online", "Find and replace", and "Console", along with various status indicators and icons for "Postbot", "Runner", "Start Proxy", "Cookies", "Vault", "Trash", and "Help".

Wiki Link

<https://github.com/htmw/2024F-Tech-Titans/wiki>

Live Demo

