IITB EdTech Internship TRACK 2
Educational Product/Application Development
(Fullstack)

Primary Report
on
Problem Identification, Validation, and Prototype Development for an Educational Application"

GROUP DETAILS

Group Name:						
Sr.	**Students Name**	**Intern_ID**	**Juno_ID**	**Dept.**	**Class**	**Roll No.**
1	Sakshi Sagar Patil		EN22166261	CSE	B.Tech	38
2	Sejal Vinod Navale		EN22	CSE	B.Tech	36
3	Mrunal Arvind Bhosale		EN22175519	CSE	B.Tech	39
4	Ayush Sunil Shirke		EN22	ENTC	B.Tech	
Faculty Mentor Name: Nandini Patil						

Section 1: Problem Identification and User Research

1.1 Identified Problem

a) SplitEasy – Smart Bill Splitting and Expense Tracker

Description:
SplitEasy is a simple, user-friendly web application designed for students and roommates living in shared accommodations. In shared living scenarios, tracking group expenses can become chaotic, often resulting in confusion, disputes, or missed payments. SplitEasy helps streamline this process by enabling users to log daily expenses, auto-calculate individual shares, and view real-time records of who owes what. It also provides features like payment reminders and monthly summaries, making it easier to manage shared bills. With its intuitive interface and collaborative features, SplitEasy promotes stress-free, organized, and transparent expense management among roommates.

Why It Matters:
Managing shared expenses manually often leads to miscommunication, missed payments, and uncomfortable financial conversations. SplitEasy addresses these issues by providing a transparent system that avoids the need for manual tracking. It ensures fair splitting of costs, supports recurring bills like rent and Wi-Fi, and encourages responsible financial behavior. By simplifying expense sharing, it not only reduces the chance of conflicts but also teaches users to develop better money management habits in shared living environments.

References:

1. **Splitwise** – A popular app that inspired the idea of shared expense tracking and fair bill splitting.
   https://www.splitwise.com
2. **Firebase** – Backend service used for real-time database and authentication in web apps like SplitEasy.
   https://firebase.google.com
3. **React** – A powerful JavaScript library used to build responsive and dynamic user interfaces.
   https://react.dev
4. **Dribbble UI Inspiration – Expense Tracker Concepts** – For UI/UX design ideas.
   https://dribbble.com/shots/14884022-Expense-Tracker-App

b) EduBridge – Offline Learning Web App for Rural Schools

Description:
EduBridge is a lightweight and inclusive web application designed to address the connectivity challenges faced by students in rural and remote areas. In many such regions, poor or intermittent internet access makes it difficult for students to benefit from digital education. EduBridge solves this problem by syncing educational content—such as videos, notes, and quizzes—whenever the internet is available, and then making all materials fully accessible offline. The app supports multilingual and interactive content, allows teachers to upload lessons, and enables students to access them anytime.

Its simple, low-data design ensures usability even in low-bandwidth environments. This empowers schools to provide consistent and quality learning experiences regardless of internet reliability.

Why It Matters:
Rural students often fall behind due to limited access to online learning resources. EduBridge addresses this gap by enabling offline learning after a one-time sync, ensuring that students can continue learning without constant internet access. It promotes educational continuity, digital inclusion, and equal learning opportunities for all. The app also reduces the reliance on manual methods like USB drives or printed materials, helping teachers manage and deliver lessons more effectively. Ultimately, EduBridge bridges the digital divide and brings modern education to underserved communities.

Reference:-

1. **Khan Academy** – A widely used educational platform offering offline access to lessons and videos via its app, similar to EduBridge's offline learning model.
   https://www.khanacademy.org
2. **Firebase** – Used for real-time syncing of data and content storage, essential for EduBridge's online-to-offline functionality.
   https://firebase.google.com
3. **React** – A JavaScript library used for building fast, responsive user interfaces, ideal for developing EduBridge's lightweight front end.
   https://react.dev
4. **Dribbble – E-Learning UI Designs** – For UI/UX design inspiration relevant to educational platforms.
   https://dribbble.com/shots/15468420-E-learning-Web-App

1.2 User Validation

a) Summary of Survey (Graph Format)

Feedback Factor	SplitEasy	EduBridge
Usefulness	9.2	9.5
Ease of Use	8.8	8.0
Real-world Need	9.4	10.0
Willingness to Use	9.1	9.6

Observation:
EduBridge scored slightly higher in real-world need and willingness**,** especially from rural education advocates. SplitEasy was also highly appreciated, especially for student and urban roommate use cases.

b) Who Was Interviewed

User Category	Count
Final-year college students (CS/IT)	6
Job seekers / prep students	4
Tech enthusiasts / indie developers	3
Healthcare users / general public	4
Total Users Interviewed	**17**

c) Visual Feedback Summary

SplitEasy – Smart Bill Splitting App

Feedback Area	Score
Usefulness	9.2
Ease of Use	8.8
Real-world Need	9.4
Willingness to Use	9.1

Remarks: Highly appreciated by students and working professionals living in shared spaces. Users found it especially helpful for transparent expense management and reducing awkward financial conversations.

EduBridge – Offline Learning App

Feedback Area	Score
Usefulness	9.5
Ease of Use	8.0
Real-world Need	10.0
Willingness to Use	9.6

Remarks: Strong appeal among rural school teachers, NGO volunteers, and education-focused users. Offline sync and multilingual support were marked as standout features for real-world impact.

Section 2: Finalized Project Idea

- **2.1 Final Problem and Solution Statement**

AptiBattle – Daily Aptitude Challenge with Friends

Problem Chosen:
Many students find it difficult to stay consistent with aptitude practice, especially when preparing for competitive exams. Traditional resources are often repetitive and lack interactive or engaging elements. There's also a shortage of multiplayer platforms that make aptitude learning enjoyable and competitive.

Why This Problem Was Selected:

- Aptitude is a core component of competitive exams like campus placements, government exams, and entrance tests.
- Most platforms offer only static quizzes with minimal engagement or motivation.
- Students are more likely to practice regularly when learning is gamified and shared with peers.

Proposed Solution:
The team will build **AptiBattle**, a web-based aptitude quiz game where users:

- Solve new aptitude questions daily across categories (quantitative, logical, reasoning)
- Compete with friends in live or timed challenges
- Track their performance on leaderboards
- Earn badges, streaks, and XP points for consistent participation
- Get instant explanations and solution breakdowns for each question

This approach makes aptitude learning fun, consistent, and peer-driven—turning routine practice into an exciting challenge.

- **2.2 Project Type:-**

  Web Application

- **2.3 Finalized Tech Stack**
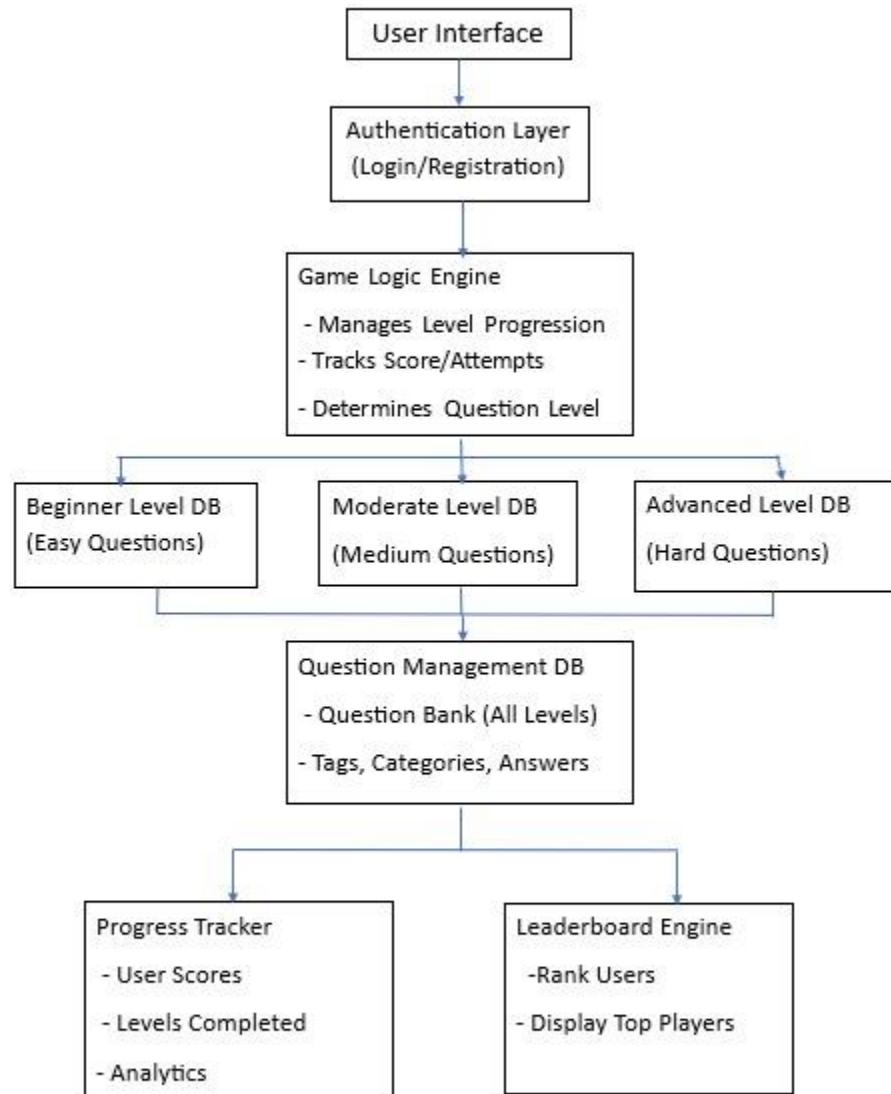
Layer	Technology
Frontend	Angular + Tailwind CSS
Backend	Spring Boot
Database	MySQL
Authentication	Firebase

2.4 Inspiration References

App Name	Notable Features	Link
Duolingo	Gamified language learning, streaks, leaderboards	duolingo.com
QuizUp (archived)	Real-time quiz battles with friends on various topics	N/A (concept reference)
WordUp	Smart vocabulary learning based on knowledge gaps	wordupapp.co

Section 3: Architecture

- **3.1 System Architecture Diagram**

```
                    ┌─────────────────┐
                    │  User Interface │
                    └─────────────────┘
                             │
                             ▼
                  ┌────────────────────────┐
                  │  Authentication Layer  │
                  │   (Login/Registration) │
                  └────────────────────────┘
                             │
                             ▼
                  ┌──────────────────────────┐
                  │  Game Logic Engine       │
                  │                          │
                  │  - Manages Level Progression │
                  │  - Tracks Score/Attempts │
                  │  - Determines Question Level │
                  └──────────────────────────┘
```

| Beginner Level DB (Easy Questions) | Moderate Level DB (Medium Questions) | Advanced Level DB (Hard Questions) |

```
                  ┌──────────────────────────┐
                  │  Question Management DB   │
                  │  - Question Bank (All Levels) │
                  │  - Tags, Categories, Answers │
                  └──────────────────────────┘
```

Progress Tracker	Leaderboard Engine
- User Scores	-Rank Users
- Levels Completed	- Display Top Players
- Analytics	

Section 4: Prototype Development

- **4.1 Feature Screens**
  - Screenshot of developed feature screen

- **4.2 GitHub Link**
  - Code repository (URL)

- **4.3 Development Challenges**

1. **Div/Card Not Centering**
   Issue: Couldn't vertically center elements.
   Solution: Used flex, items-center, justify-center, min-h-screen.
2. **Tailwind Styles Not Applying**
   Issues: Classes not rendering.
   Solution: Verified tailwind.config.js, used @tailwind directives in main CSS.
3. **PostCSS Configuration Errors**
   Issue: Missing or incorrect plugins.
   Solution: Installed tailwindcss, autoprefixer, and added them to postcss.config.js.
4. **Changes Not Reflecting**
   Issue: Tailwind not updating in dev build.
   Solution: Fixed file paths in tailwind.config.js > content and restarted server.

Section 5: Conclusion

Summary of Progress Made

The team has finalized the concept and technical groundwork for **AptiBattle**—a web-based, multiplayer aptitude quiz game designed to make logical and quantitative practice more interactive. The initial phase included problem identification, user validation, and selection of the tech stack: **React.js** for the frontend, **Spring Boot** for the backend, and **MySQL** for database management. A

basic UI structure has been developed, including the **host dashboard**, **player interface**, and **real-time leaderboard**. Tailwind CSS has been successfully integrated with PostCSS, and initial styling issues like card alignment and layout centering were resolved. Key functionality like game code generation and multiplayer flow has been outlined in the planning phase.

Next Steps in Development

The team will now focus on implementing the core game mechanics. The **host panel** will allow quiz creators to upload or create aptitude questions and generate a unique code for the session. Players will use this code to join the game, view questions, and compete in real time. Socket-based communication will be integrated to handle live multiplayer gameplay and leaderboard updates. Additional features like **user authentication**, **progress tracking**, **badges**, and **friend battles** will be developed. The UI will be optimized for mobile users, and thorough testing will be conducted to ensure a seamless multiplayer experience.