Neural Breach - Hackathon

Yugastr 2026

These features are **MANDATORY** and must be implemented during the event. They constitute **35%** of your total evaluation.

1. User Authentication & Profiles

Requirements:

• **Registration & Login System:** Users can create accounts and login (use any authentication method - email/password, phone, or social login)

• **User Profile:** Each user must have a profile containing:
- Name
- College/Institution name
- Branch/Department
- Current semester/year
- Optional: Profile picture, bio

• **Session Management:** Keep users logged in across sessions

2. Resource Upload & Management

Requirements:

• **File Upload:** Allow users to upload files (PDF, DOCX, images, PPT, etc.)

• **Resource Information:** When uploading, users must provide:
- Resource title
- Subject/Course name
- Semester
- Resource type (Notes, Question Papers, Solutions, Project Reports, Study Material)
- Year/Batch
- Optional description

• **Tags/Keywords:** Allow users to add searchable tags (e.g., 'data-structures', 'algorithms', 'mid-term')

• **Edit/Delete:** Users can edit details or delete their own uploaded resources

• **File Storage:** Implement proper file storage (local filesystem or cloud storage like AWS S3, Cloudinary, etc.)

3. Access Control & Privacy Settings

Requirements:

• **Private Access:** Resources marked 'Private' can only be accessed by users from the SAME college/institution

• **Public Access:** Resources marked 'Public' can be accessed by users from ANY college/institution

• **Default Setting:** Allow uploader to choose privacy level during upload

• **Access Verification:** System must verify user's college before granting access to private resources

• **Visual Indicators:** Clearly show which resources are Private vs Public

> **Example:** A student from 'XYZ College' uploads Data Structures notes and marks them as 'Private'. Only students who registered with 'XYZ College' in their profile can view and download these notes. If marked 'Public', students from any college can access them.

4. Search & Filter System

Requirements:

• **Search Functionality:** Search by resource title, subject name, tags, or keywords

• **Filter Options:** Users must be able to filter by:
- Subject/Course
- Semester
- Resource type (Notes, Question Papers, etc.)
- Branch/Department
- Year/Batch
- Privacy level (Public/Private)

• **Sort Options:** Sort results by:
- Latest uploads (most recent first)
- Highest rated
- Most popular

• **Combined Filters:** Allow multiple filters to work together (e.g., 'Semester 3' + 'Computer Science' + 'Notes')

5. Rating & Review System

Requirements:

• **Star Rating:** Users can rate resources on a scale of 1-5 stars

• **Written Reviews:** Users can write text reviews/comments about the resource quality

• **Review Display:** Show all reviews for each resource

• **One Review Per User:** Each user can only submit one review per resource (but can edit it later)

• **Rating Summary:** Display average rating on resource cards/listings

TECHNICAL REQUIREMENTS

- **Frontend:** Clean, responsive UI that works on both desktop and mobile devices
- **Backend:** RESTful API with proper endpoints for all features
- **Database:** Properly designed database schema with relationships between users, resources, reviews, etc.
- **File Handling:** Secure file upload and storage mechanism
- **Authentication:** Secure user authentication and session management
- **Authorization:** Proper access control - users should only see/access resources they're allowed to
- **Input Validation:** Validate all user inputs to prevent security issues
- **Error Handling:** Proper error messages and handling throughout the application

UI/UX REQUIREMENTS

- **Resource Cards/Listings:** Display resources in an organized grid or list view showing key information
- **Resource Detail Page:** Clicking a resource should show full details, reviews, and download option
- **User Dashboard:** Show user's uploaded resources and activity
- **Navigation:** Clear navigation menu to access different sections (Home, Search, Upload, Profile)
- **Responsive Design:** Interface should adapt to different screen sizes
- **Visual Feedback:** Show loading states, success/error messages, confirmations

EXPECTED USER FLOWS

Flow 1: New User Registration & Upload

User registers → Fills profile (name, college, branch, semester)

User logs in → Goes to Upload section

Uploads file → Fills details (title, subject, type, tags)

Selects privacy: Private or Public → Submits

Resource appears in their profile and in search results

Flow 2: Searching & Downloading

User logs in → Goes to Search/Browse section

Applies filters (subject: Data Structures, semester: 3, type: Notes)

Views results → Clicks on a resource card

Sees full details, reviews, rating → Downloads the file

Optionally leaves a rating and review

Flow 3: Access Control

Student from 'ABC College' tries to access a Private resource from 'XYZ College'

System checks: User's college ≠ Resource's college → Access Denied

Shows message: 'This resource is private and only available to XYZ College students'

Student can still access Public resources from any college

EVALUATION BREAKDOWN

Feature	Weight
User Authentication & Profiles	20%
Resource Upload & Management	25%
Access Control & Privacy Settings	20%
Search & Filter System	20%
Rating & Review System	15%

Note: These mandatory features account for 35% of total marks. The remaining 65% comes from Code Quality (25%), Innovation & UX (25%), and Live Demo & Q&A; (15%).

IMPORTANT NOTES

• All features must be implemented and functional during the live demo

• Focus on core functionality first, then improve UI/UX

• Use dummy/sample data if needed for demonstration

• Test all user flows before the presentation

• Be prepared to explain your implementation during Q&A;

• Additional innovative features beyond mandatory ones will earn bonus points

Time Allocated: 5.5 Hours | Start Time: 11:00 AM | Good Luck! ∎