EDUNEXUS

Stack Holders :-

→ Students
→ Admin
→ Faculty
→ Teaching Assistants

→ login/signup page with cookies and Capcha & 2-step authentication for signup & encrypted password

Customred Dashboards

• Student Administration System:

→ Fees Payment
  hostel alloatment
  Course registration
• If Student Pay the fees
  then fee reciept is generated
  but due to some problem
  reciept is not generated then
  what can Student do?

Course registration:
→ Student can't enroll Course same
  more than twice.
→ If a student select one
  Course then all other Course
  which are in same slot are
  hidden.

Functional Requirements :~

↪ Authentication & Authorization
↪ Dashboard
↪ Profile
↪ Notifications & Reminders
↪ Upcoming events
↪ Global chatbot
↪ Private chatbot
↪ Course registration
↪ Test section
↪ Report & Grade card
↪ Feedback & Help support

⇒ Daily Reminder

→ These are
  lectures you have
  to attend at
  this place...
→ your exam is
  next week
→ you have
  assignment due
  date.

Idea:1 Providing Student's Profile

Non-functional Requirements :~

↪ Scalability
↪ Usability
↪ Security
↪ User friendly
↪ Availability
↪ Performance
↪ User satisfaction
↪ Reliability
↪ Compliance
↪ Audibility

make arrangements for the
professor to take exams/quizs
of any kind of test activity

* Communication tools:-
  facility for messaging, emails,
  announcements, discussion
  forums to communicate with
  professors, TAs, batchmates, peers.

Roles :-

→ Frontend developer
→ Backend developer
→ UI/UX designer
→ Documentation
→ Database developer
→ Tester
→ System designer

* storage for adding personal
  additional notes (hand-written
  or any format) for each course/
  lab in each semester, which
  can be stored till an individual
  gets graduated.

Tools & Tech Stack :~

→ React
→ NodeJS
→ ExpressJS
→ MongoDB
→ Figma

* Help or support feature:-
  Any help or advice if anyone
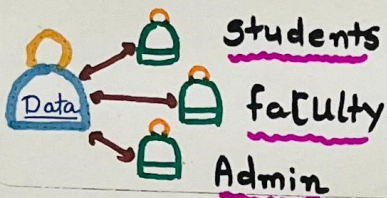 face at any stage in website
  or system handling.

WHAT IS THE CONCEPT CALLED?

EDUNEXUS

- A comprehensive platform that centralized all institutional academic related services.

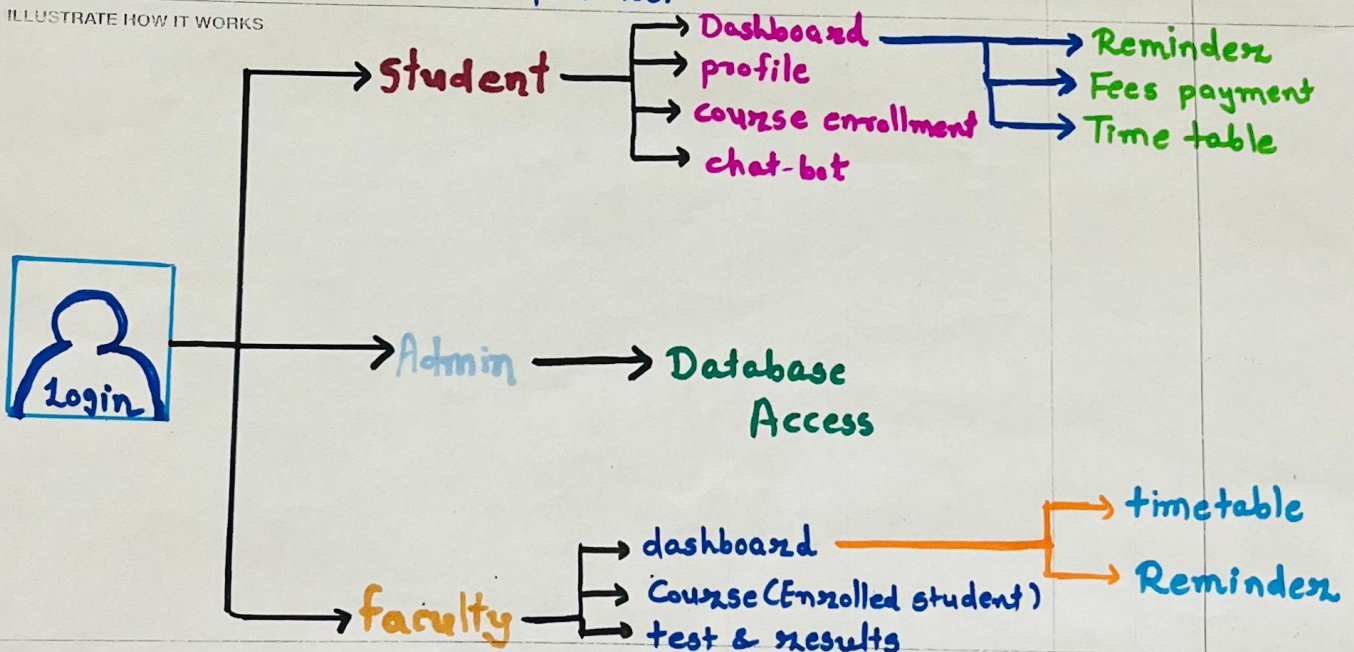**WHO IS IT FOR?**

Data → Students
faculty
Admin

WHAT PROBLEM DOES IT SOLVE?

It solves the problem of fragmentation of services, Inefficiently and time consumption communication gaps. Accessbility & user experience.

WHAT IS THE BIG IDEA?

CENTRALIZED STUDENT SERVICE

ILLUSTRATE HOW IT WORKS

Login →

Student →
Dashboard
profile → Reminder
course enrollment → Fees payment
chat-bot → Time table

Admin → Database Access

faculty →
dashboard
Course (Enrolled student) → timetable
test & results → Reminder

WHY MIGHT IT FAIL?

- Resistance to technological adoptation
- lack of Feature Utilization

WHAT SHOULD WE PROTOTYPE AND TEST?

- User Interface (UI) prototype
- Data collection & Management
- Actor feedback

HOW MIGHT WE MEASURE SUCCESS?

- User engagement
- Adoptation Rate
- System feedback
- Platform ability to improve outcomes

HOW WILL WE MAKE THIS HAPPEN?

Requirements
UI/UX prototype
development Backend
Documentation Maintenance

Design & Analysis
Foontent development
Testing
Deployment