

Database Project Proposal

International Student Buddy System

1. Database Purpose: What Problem Are We Solving?

Background & Problem Statement

International students face challenges adjusting to a new academic and social environment due to cultural differences, language barriers, and unfamiliarity with university resources. Universities provide support programs, but there is no structured, data-driven way to match new students with experienced mentors who can guide them through this transition.

This project proposes a database-driven Buddy System, where new international students are paired with experienced students (mentors) using a matching algorithm based on:

- Nationality & Language – Ensuring new students can communicate comfortably.
- Academic Program – If no nationality or language match is found, pairing is based on course similarity for academic guidance.

Target Audience & Users

- New International Students – Need guidance in academics, culture, housing, and campus life.
- Experienced Students (Buddies/Mentors) – Willing to guide new students based on shared experiences.
- University Student Support Services – To track mentor-mentee relationships and improve student assistance programs.

Key Benefits

- Easier transition for new international students.
- Stronger community connections for students from similar backgrounds.
- Better access to campus resources through structured guidance.

2. Database Schema: Tables & Relationships

The database consists of **four main tables** to track students, mentors, matches, and university resources.

Entity-Relationship Diagram (ERD)

Students (Student_ID)

↓ (FK)

Matches (Match_ID)

↓ (FK)

Buddies (Buddy_ID)

↓ (FK)

Resources (Resource_ID)

Table Descriptions & Keys

Table Name	Attributes	Primary Key (PK)	Foreign Key (FK)
Students	Student_ID, Name, Email, Country, Language, Program, Arrival_Date	Student_ID	—
Buddies	Buddy_ID, Name, Email, Country, Language, Expertise, Program	Buddy_ID	—
Matches	Match_ID, Student_ID, Buddy_ID, Status (Pending, Matched, Completed)	Match_ID	Student_ID (FK), Buddy_ID (FK)
Resources	Resource_ID, Type (Housing, Financial Aid, Counselling), Contact_Info	Resource_ID	—

How the Tables Relate:

- Students table stores new international students seeking mentorship.
- Buddies table stores mentors available to assist students.
- Matches table links students and mentors based on nationality, language, or academic program.
- Resources table provides access to key support services like housing, financial aid, and visa help.

3. Interface Design: How Users Will Interact

- If expanded into a web-based database system, the following interfaces would be designed:
- Student Registration Page – New students input their country, language, and academic program.
- Mentor Directory – Students can search for buddies based on nationality, language, or program.
- Resource Portal – A categorized directory of campus support services.

Interface Sketch Example

User Role	Main Features
New Student (Mentee)	Register, Find a Mentor, Access Resources

Mentor (Buddy)	Register, View Matched Students, Update Availability
Admin (University Staff)	Monitor Matches, Manage Student Feedback, Update Resources

4. Data Collection: Sources of Information

Since actual student data is confidential, I will use:

- Dummy Data – Manually created sample student and mentor profiles.
- University Websites – For public campus resource details.
- Simulated Survey Data – If feasible, conduct a sample survey to mimic real student responses.

5. Methods: SQL Queries & Data Processing

The project will use basic SQL queries for mentor-student matching and data retrieval.

1. Primary Matching – Nationality and Language:

```
SELECT b.*
FROM Buddies b
JOIN Students s ON b.Country = s.Country OR b.Language = s.Language
WHERE s.Student_ID = 1; -- Find a buddy for Student_ID 1
```

2. Secondary Matching – If No Match by Nationality, Match by Academic Program:

```
SELECT b.*
FROM Buddies b
JOIN Students s ON b.Program = s.Program
WHERE s.Student_ID = 1
AND b.Buddy_ID NOT IN (
SELECT Buddy_ID FROM Matches WHERE Student_ID = 1
```

3. Finding Students Without a Mentor Yet:

```
SELECT s.Name FROM Students s
LEFT JOIN Matches m ON s.Student_ID = m.Student_ID
WHERE m.Buddy_ID IS NULL;
```