***IamOnCampus DB Codes***

CREATE TABLE Users (

User\_id INT NOT NULL AUTO\_INCREMENT,

F\_name VARCHAR(30) NOT NULL,

M\_name VARCHAR(30),

L\_name VARCHAR(30) NOT NULL,

Passwords VARCHAR(255) NOT NULL,

Role VARCHAR(50) NOT NULL,

DeptName VARCHAR(100) NOT NULL,

PRIMARY KEY(User\_id));

ALTER TABLE Users ADD COLUMN about\_me VARCHAR(500) NOT NULL;

CREATE TABLE User\_Emails (

User\_id INT NOT NULL,

Email VARCHAR(100) NOT NULL,

PRIMARY KEY (User\_id, Email),

FOREIGN KEY (User\_id) REFERENCES Users(User\_id)

ON UPDATE CASCADE ON DELETE CASCADE);

CREATE TABLE Skills (

Skill\_id INT NOT NULL AUTO\_INCREMENT,

Skill\_name VARCHAR(100) NOT NULL,

Skill\_description VARCHAR(500),

Availability\_time VARCHAR(50),

Mode VARCHAR(50),

PRIMARY KEY (Skill\_id));

CREATE TABLE Skill\_Requests (

Request\_id INT NOT NULL AUTO\_INCREMENT,

Skill\_id INT NOT NULL,

User\_id INT NOT NULL,

PRIMARY KEY (Request\_id),

FOREIGN KEY (Skill\_id) REFERENCES Skills(Skill\_id),

FOREIGN KEY (User\_id) REFERENCES Users(User\_id)

ON UPDATE CASCADE ON DELETE CASCADE);

CREATE TABLE Notifications (

Notify\_id INT NOT NULL AUTO\_INCREMENT,

Seen\_status BOOLEAN NOT NULL DEFAULT FALSE,

Message VARCHAR(500) NOT NULL,

User\_id INT NOT NULL,

PRIMARY KEY (Notify\_id),

FOREIGN KEY (User\_id) REFERENCES Users(User\_id)

ON UPDATE CASCADE

ON DELETE CASCADE);

CREATE TABLE Login\_History (

Login\_id INT NOT NULL AUTO\_INCREMENT,

Login\_time DATETIME NOT NULL,

User\_id INT NOT NULL,

PRIMARY KEY (Login\_id),

FOREIGN KEY (User\_id) REFERENCES Users(User\_id)

ON UPDATE CASCADE

ON DELETE CASCADE);

CREATE TABLE Academic\_Posts (

Post\_id INT NOT NULL AUTO\_INCREMENT,

Acd\_title VARCHAR(255) NOT NULL,

Date\_posted DATETIME NOT NULL,

Acd\_content VARCHAR(2000) NOT NULL,

Course VARCHAR(100),

User\_id INT NOT NULL,

PRIMARY KEY (Post\_id),

FOREIGN KEY (User\_id) REFERENCES Users(User\_id)

ON UPDATE CASCADE

ON DELETE CASCADE);

CREATE TABLE Replies (

Acd\_reply\_id INT NOT NULL AUTO\_INCREMENT,

Reply\_content VARCHAR(1000) NOT NULL,

Reply\_date DATETIME NOT NULL,

Post\_id INT NOT NULL,

PRIMARY KEY (Acd\_reply\_id),

FOREIGN KEY (Post\_id) REFERENCES Academic\_Posts(Post\_id)

ON UPDATE CASCADE

ON DELETE CASCADE);

CREATE TABLE Tags (

Tag\_id INT NOT NULL AUTO\_INCREMENT,

Tag\_name VARCHAR(100) NOT NULL,

Post\_id INT NOT NULL,

PRIMARY KEY (Tag\_id),

FOREIGN KEY (Post\_id) REFERENCES Academic\_Posts(Post\_id)

ON UPDATE CASCADE

ON DELETE CASCADE);

CREATE TABLE Club (

Club\_id INT NOT NULL AUTO\_INCREMENT,

Club\_name VARCHAR(100) NOT NULL,

PRIMARY KEY (Club\_id));

CREATE TABLE Club\_Membership (

Joining\_date DATE NOT NULL,

Club\_id INT NOT NULL,

User\_id INT NOT NULL,

PRIMARY KEY (Club\_id, User\_id),

FOREIGN KEY (Club\_id) REFERENCES Club(Club\_id)

ON UPDATE CASCADE

ON DELETE CASCADE,

FOREIGN KEY (User\_id) REFERENCES Users(User\_id)

ON UPDATE CASCADE

ON DELETE CASCADE);

CREATE TABLE Events (

Event\_id INT NOT NULL AUTO\_INCREMENT,

Event\_title VARCHAR(255) NOT NULL,

Event\_date DATETIME NOT NULL,

Event\_location VARCHAR(255) NOT NULL,

Event\_description VARCHAR(1000),

Club\_id INT NOT NULL,

PRIMARY KEY (Event\_id),

FOREIGN KEY (Club\_id) REFERENCES Club(Club\_id)

ON UPDATE CASCADE

ON DELETE CASCADE);

CREATE TABLE Event\_Participants (

Participant\_id INT NOT NULL AUTO\_INCREMENT,

Phone VARCHAR(20),

User\_id INT NOT NULL,

PRIMARY KEY (Participant\_id),

FOREIGN KEY (User\_id) REFERENCES Users(User\_id)

ON UPDATE CASCADE

ON DELETE CASCADE);

CREATE TABLE Event\_Registration (

Event\_id INT NOT NULL,

Participant\_id INT NOT NULL,

PRIMARY KEY (Event\_id, Participant\_id),

FOREIGN KEY (Event\_id) REFERENCES Events(Event\_id)

ON UPDATE CASCADE

ON DELETE CASCADE,

FOREIGN KEY (Participant\_id) REFERENCES Event\_Participants(Participant\_id)

ON UPDATE CASCADE

ON DELETE CASCADE);

CREATE TABLE Discussion\_Threads (

Thread\_id INT NOT NULL AUTO\_INCREMENT,

Dt\_title VARCHAR(255) NOT NULL,

Dt\_content VARCHAR(2000) NOT NULL,

Dt\_tag VARCHAR(100),

Timestamp DATETIME NOT NULL,

User\_id INT NOT NULL,

PRIMARY KEY (Thread\_id),

FOREIGN KEY (User\_id) REFERENCES Users(User\_id)

ON UPDATE CASCADE

ON DELETE CASCADE);

CREATE TABLE Thread\_Replies (

Reply\_id INT NOT NULL AUTO\_INCREMENT,

Treply\_content VARCHAR(1000) NOT NULL,

Treply\_date DATETIME NOT NULL,

Thread\_id INT NOT NULL,

PRIMARY KEY (Reply\_id),

FOREIGN KEY (Thread\_id) REFERENCES Discussion\_Threads(Thread\_id)

ON UPDATE CASCADE

ON DELETE CASCADE);

CREATE TABLE Profiles (

Profile\_id INT AUTO\_INCREMENT PRIMARY KEY,

User\_id INT NOT NULL,

DP VARCHAR(255), -- Path to uploaded image

Bio TEXT, -- Optional short description or status

FOREIGN KEY (User\_id) REFERENCES Users(User\_id)

);

CREATE TABLE UserPosts (

Post\_id INT AUTO\_INCREMENT PRIMARY KEY,

User\_id INT NOT NULL,

Content TEXT NOT NULL,

Created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (User\_id) REFERENCES Users(User\_id) ON DELETE CASCADE

);

CREATE TABLE post\_reactions(

`Reaction\_id` INT AUTO\_INCREMENT PRIMARY KEY,

`Post\_id` INT NOT NULL,

`User\_id` INT NOT NULL,

`Reaction\_type` VARCHAR(20) NOT NULL,

`Timestamp` DATETIME DEFAULT CURRENT\_TIMESTAMP,

UNIQUE(Post\_id, User\_id), -- Each user can react once per post

FOREIGN KEY (`Post\_id`) REFERENCES `userposts`(`Post\_id`) ON DELETE CASCADE,

FOREIGN KEY (`User\_id`) REFERENCES `users`(`User\_id`) ON DELETE CASCADE

);

CREATE TABLE post\_comments (

`Comment\_id` INT AUTO\_INCREMENT PRIMARY KEY,

`Post\_id` INT NOT NULL,

`User\_id` INT NOT NULL,

`Comment\_text` TEXT NOT NULL,

`Timestamp` DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (`Post\_id`) REFERENCES `userposts`(`Post\_id`) ON DELETE CASCADE,

FOREIGN KEY (`User\_id`) REFERENCES `users`(`User\_id`) ON DELETE CASCADE

);

CREATE TABLE messages (

Message\_id INT AUTO\_INCREMENT PRIMARY KEY,

Sender\_id INT NOT NULL,

Receiver\_id INT NOT NULL,

Message\_text TEXT NOT NULL,

Timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP,

Seen\_status TINYINT(1) DEFAULT 0,

FOREIGN KEY (Sender\_id) REFERENCES users(User\_id) ON DELETE CASCADE,

FOREIGN KEY (Receiver\_id) REFERENCES users(User\_id) ON DELETE CASCADE

);