DATABASE TABLE DESIGN – CAMPUSLANCE

Table: USERS

Primary Key: id

Field Name	Data Type(size)	Key Constraints	Description
id	INT	PRIMARY KEY	Unique identifier for each user
name	VARCHAR(150)	—	Full name of the user
email	VARCHAR(200)	UNIQUE	Login email address
password_hash	VARCHAR(255)	—	Encrypted password
role	ENUM('freelancer','client','admin')	—	User role
created_at	DATETIME	—	Account creation time
updated_at	DATETIME	—	Last update time

Table: FREELANCER_PROFILES

Primary Key: user_id

Foreign Key: user_id->id(users)

Field Name	Data Type(Size)	Key Constraints	Description
user_id	INT	PRIMARY KEY, FOREIGN KEY → users(id)	Freelancer account
bio	TEXT	—	Freelancer description
skills	VARCHAR(255)	—	Skill set

Field Name	Data Type(Size)	Key Constraints	Description
skill_score	DECIMAL(4,2)	—	Verified skill score
availability	VARCHAR(50)	—	Availability status

Table: GIGS

Primary Key: id

Foreign Key: freelancer_id - >id(users)

Field Name	Data Type(size)	Key Constraints	Description
id	INT	PRIMARY KEY	Gig identifier
freelancer_id	INT	FOREIGN KEY → users(id)	Gig owner
title	VARCHAR(200)	—	Service title
description	TEXT	—	Service details
price	DECIMAL(10,2)	—	Service price
delivery_days	INT	—	Delivery time
tags	VARCHAR(255)	—	Searchable tags
status	ENUM('active','paused','removed')	—	Gig status
created_at	DATETIME	—	Created time
updated_at	DATETIME	—	Last updated

Table: GIG_TEMPLATES

Primary Key: id

Foreign Key: freelancer_id->id(users)

Field Name	Data Type(size)	Key Constraints	Description
id	INT	PRIMARY KEY	Template ID
freelancer_id	INT	FOREIGN KEY → users(id)	Template owner
name	VARCHAR(150)	—	Template name
preset_title	VARCHAR(200)	—	Default title
preset_description	TEXT	—	Default description
preset_price	DECIMAL(10,2)	—	Default price
created_at	DATETIME	—	Created time

Table: REQUIREMENTS

Primary Key: id

Foreign Key: client_id->id(users)

Field Name	Data Type(size)	Key Constraints	Description
id	INT	PRIMARY KEY	Requirement ID
client_id	INT	FOREIGN KEY → users(id)	Client
title	VARCHAR(200)	—	Requirement title
description	TEXT	—	Requirement details
tags	VARCHAR(255)	—	Searchable tags
deadline	DATETIME	—	Expected completion
created_at	DATETIME	—	Posted time

Table: BIDS

Primary Key: id

Foreign Key: requirement_id ->id(requirements) ; freelancer_id ->id(users)

Field Name	Data Type(size)	Key Constraints	Description
id	INT	PRIMARY KEY	Bid identifier
requirement_id	INT	FOREIGN KEY → requirements(id)	Requirement
freelancer_id	INT	FOREIGN KEY → users(id)	Freelancer
price	DECIMAL(10,2)	—	Quoted price
delivery_days	INT	—	Proposed duration
message	TEXT	—	Bid message
created_at	DATETIME	—	Bid time

Table: ORDERS

Primary Key: id

Foreign Key: requirement_id->id(requirements);gig_id->id(gigs);client_id->id(users)

freelancer_id->id(users);bid_id->id(bids)

Field Name	Data Type(size)	Key Constraints	Description
id	INT	PRIMARY KEY	Order ID
requirement_id	INT	FOREIGN KEY → requirements(id)	From requirement
gig_id	INT	FOREIGN KEY → gigs(id)	Direct gig order
client_id	INT	FOREIGN KEY → users(id)	Client

Field Name	Data Type(size)	Key Constraints	Description
freelancer_id	INT	FOREIGN KEY → users(id)	Freelancer
bid_id	INT	FOREIGN KEY → bids(id)	Accepted bid
status	ENUM('created','in_progress','revision', 'completed','cancelled','disputed')	—	Order status
total_price	DECIMAL(10,2)	—	Final price
created_at	DATETIME	—	Order time
updated_at	DATETIME	—	Last update

Table: MILESTONES

Primary Key: id

Foreign Key: order_id->id(orders)

Field Name	Data Type(size)	Key Constraints	Description
id	INT	PRIMARY KEY	Milestone ID
order_id	INT	FOREIGN KEY → orders(id)	Order
title	VARCHAR(150)	—	Milestone title
description	TEXT	—	Details
due_date	DATETIME	—	Due date

Field Name	Data Type(size)	Key Constraints	Description
status	ENUM('pending','submitted','approved','rejected')	—	Milestone state
sequence	INT	—	Execution order

Table: REVISIONS

Primary Key: id

Foreign Key: order_id->id(orders); milestone_id->id(milestones);requested_by->id(users)

Field Name	Data Type	Key Constraints	Description
id	INT	PRIMARY KEY	Revision ID
order_id	INT	FOREIGN KEY → orders(id)	Order
milestone_id	INT	FOREIGN KEY → milestones(id)	Milestone
requested_by	INT	FOREIGN KEY → users(id)	Requester
comments	TEXT	—	Revision notes
revision_number	INT	—	Revision count
created_at	DATETIME	—	Request time

Table: FILES

Primary Key: id

Foreign Key: id(users)

Field Name	Data Type(size)	Key Constraints	Description
id	INT	PRIMARY KEY	File ID

Field Name	Data Type(size)	Key Constraints	Description
owner_id	INT	FOREIGN KEY → users(id)	Uploaded by
file_name	VARCHAR(255)	—	File name
file_path	VARCHAR(500)	—	Storage path
mime_type	VARCHAR(100)	—	File type
size	BIGINT	—	File size
created_at	DATETIME	—	Upload time

Table: PORTFOLIO_ITEMS

Primary Key: id

Foreign Key: freelancer_id->id(users);file_id->id(files)

Field Name	Data Type	Key Constraints	Description
id	INT	PRIMARY KEY	Portfolio ID
freelancer_id	INT	FOREIGN KEY → users(id)	Owner
title	VARCHAR(200)	—	Work title
category	VARCHAR(100)	—	Category
description	TEXT	—	Details
file_id	INT	FOREIGN KEY → files(id)	Linked file
created_at	DATETIME	—	Added time

Table: MESSAGES

Primary Key: id

Foreign Key: order_id->id(orders);sender_id->id(users);file_id->id(files)

Field Name	Data Type	Key Constraints	Description
id	INT	PRIMARY KEY	Message ID
order_id	INT	FOREIGN KEY → orders(id)	Order
sender_id	INT	FOREIGN KEY → users(id)	Sender
text	TEXT	—	Message content
file_id	INT	FOREIGN KEY → files(id)	Attachment
sent_date	DATE	—	Date sent
sent_time	TIME	—	Time sent

Table: RATINGS

Primary Key: id

Foreign Key: order_id->id(orders);client_id->id(users);freelancer_id->id(users)

Field Name	Data Type	Key Constraints	Description
id	INT	PRIMARY KEY	Rating ID
order_id	INT	FOREIGN KEY → orders(id)	Order
client_id	INT	FOREIGN KEY → users(id)	Reviewer
freelancer_id	INT	FOREIGN KEY → users(id)	Rated freelancer
stars	INT	CHECK (1–5)	Rating value

Field Name	Data Type	Key Constraints	Description
comment	TEXT	—	Feedback
created_at	DATETIME	—	Rating time

Table: NOTIFICATIONS

Primary Key: id

Foreign Key: user_id->id(users)

Field Name	Data Type(size)	Key Constraints	Description
id	INT	PRIMARY KEY	Notification ID
user_id	INT	FOREIGN KEY → users(id)	Receiver
type	VARCHAR(100)	—	Notification type
payload	JSON	—	Extra data
is_read	BOOLEAN	—	Read status
created_at	DATETIME	—	Sent time

Table: FAVORITES

Primary Key: id

Foreign Key: user_id->id(users)

Field Name	Data Type	Key Constraints	Description
id	INT	PRIMARY KEY	Favorite ID
user_id	INT	FOREIGN KEY → users(id)	User
fav_type	ENUM('gig','freelancer')	—	Type

Field Name	Data Type	Key Constraints	Description
target_id	INT	—	Target ID
created_at	DATETIME	—	Added time

Normalization reasoning (1NF → 3NF)

1NF (First Normal Form)

- All tables have atomic values: each column contains a single value (no repeating groups).
- Example: gigs stores one price per row, not a list of prices.

2NF (Second Normal Form)

- Achieved because all non-key attributes are fully functionally dependent on the primary key.
- Example: In gigs, attributes like title, price, delivery_days depend on gig.id (the PK). We separated tags into tags + gig_tags to avoid partial dependencies.

3NF (Third Normal Form)

- Achieved by removing transitive dependencies.
- Example: files table stores file metadata once. portfolio_items and messages reference files via file_id rather than duplicating filename/mime/size.
- freelancer_profiles separated from users to avoid storing many profile-specific fields in the users table.