***Folder Structure***

***mini-twitter-clone***

**backend**/

**frontend**/

── **public/**

**── src/**

── **components**/

──Login/Register

── dashboard

── PostList.js

── PostForm.js

── CommentList.js

── CommentForm.js

── **services**/

── api.js

── **App.js**

── **App.css**

**── index.js**

**── package. Json**

**── README.md**

── **config**/

── database.js

── **controllers**/

──userController.js

── postController.js

── commentController.js

── **middleware**/

── validation.js

── **models**/

── Users.js

── Post.js

── Comment.js

── **routes**/

── users.js

── posts.js

── comments.js

── **utils**/

── sanitizer.js

**── package. Json**

**── server.js**

**TABLES**

**User**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User\_Id** | **User\_name** | **User\_email** | **User\_password** | **Created\_at** |
|  |  |  |  |  |

**Post**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Posts\_Id** | **User\_id** | **Post\_content** | **Created\_at** | **Updated\_at** |

**Comments**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Comments\_id** | **Post\_id** | **User\_id** | **Comment** | **Created\_at** | **Updated\_at** |

**TABLE SCHEMA**

CREATE TABLE user (

user\_id SERIAL PRIMARY KEY,

user\_name VARCHAR(50) NOT NULL CHECK (char\_length(name) >= 2 AND char\_length(name) <= 50),

user\_email VARCHAR(100) NOT NULL UNIQUE CHECK (position(' ' in email) = 0),

user\_password VARCHAR(255) NOT NULL, -- Store hashed passwords

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**-- Add a description for the table**

**COMMENT ON TABLE users IS 'Stores registered users who can create posts and comments';**

**-- Add column-level comments**

**COMMENT ON COLUMN user.id IS 'Unique identifier for each user (auto-incremented)';**

**COMMENT ON COLUMN user.user\_name IS 'Name of the user (2–50 characters)';**

**COMMENT ON COLUMN user.user\_email IS 'Unique email address used for user login (no spaces allowed)';**

**COMMENT ON COLUMN user.user\_password IS 'User''s password stored in hashed form using bcrypt or similar';**

**COMMENT ON COLUMN user.created\_at IS 'Timestamp when the user account was created';**

-- POST TABLE

CREATE TABLE post (

post\_id SERIAL PRIMARY KEY,

user\_id INT NOT NULL REFERENCES users(id) ON DELETE CASCADE,

post\_content TEXT NOT NULL CHECK (char\_length(content) >= 5 AND char\_length(content) <= 250),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**-- Add a description for the table**

**COMMENT ON TABLE post IS 'Stores posts (tweets) created by users';**

**-- Add comments for each column**

**COMMENT ON COLUMN post.post\_id IS 'Unique identifier for each post (auto-incremented)';**

**COMMENT ON COLUMN post.user\_id IS 'References the user who created the post';**

**COMMENT ON COLUMN post.post\_content IS 'Main content of the post (between 5 and 250 characters)';**

**COMMENT ON COLUMN post.created\_at IS 'Timestamp when the post was created';**

**COMMENT ON COLUMN post.updated\_at IS 'Timestamp when the post was last updated';**

-- Create the comment table

CREATE TABLE comment (

comment\_id SERIAL PRIMARY KEY,

post\_id INT NOT NULL REFERENCES post(post\_id) ON DELETE CASCADE,

user\_id INT NOT NULL REFERENCES users(id) ON DELETE CASCADE,

comment TEXT NOT NULL CHECK (char\_length(comment) >= 2 AND char\_length(comment) <= 200),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**-- Add a description for the table**

**COMMENT ON TABLE comment IS 'Stores user comments made on posts';**

**-- Add comments for each column**

**COMMENT ON COLUMN comment.comment\_id IS 'Unique identifier for each comment (auto-incremented)';**

**COMMENT ON COLUMN comment.post\_id IS 'References the post to which this comment belongs';**

**COMMENT ON COLUMN comment.user\_id IS 'References the user who made the comment';**

**COMMENT ON COLUMN comment.comment IS 'Text content of the comment (between 2 and 200 characters)';**

**COMMENT ON COLUMN comment.created\_at IS 'Timestamp when the comment was created';**

**COMMENT ON COLUMN comment.updated\_at IS 'Timestamp when the comment was last updated';**