

Code Structure

The application is structured as follows:

- **app/**: Contains the core application code.
 - **Http/**: Contains HTTP-related code.
 - **Controllers/**: Contains the controllers that handle HTTP requests.
 - **CommentController.php**: Handles CRUD operations for comments.
 - **ReplyController.php**: Handles CRUD operations for replies.
 - **Requests/**: Contains form request validation classes.
 - **Resources/**: Contains API resource classes for transforming models.
 - **Models/**: Contains Eloquent models representing database tables.
- **database/**: Contains database migrations, factories, and seeders.
- **public/**: Contains publicly accessible files like CSS, JS, and images.
- **resources/**: Contains view files and frontend assets.
- **routes/**: Contains route definitions.

Important Functions

1- Located at [CommentController.php](#).

- **index()**: Retrieves all comments along with their replies, ordered by creation date in descending order.

```
public function index()
{
    $comments = Comment::with('replies')->orderBy('created_at', 'desc')->get();

    return $this->apiResponse(CommentResource::collection($comments));
}
```

- **show(\$id)**: Retrieves a specific comment by its ID along with its replies.

```

public function show($id)
{
    $comment = Comment::find($id);

    if(!$comment){
        return response()->json([
            'error' => 'Comment Not Found',
            'code' => 404,
        ]);
    }

    $comment->load('replies');

    return $this->apiResponse(new CommentResoucre($comment));
}

```

- **store(StoreCommentRequest \$request)**: Stores a new comment.

```

public function store(StoreCommentRequest $request)
{
    $validated = $request->validated();
    $comment = Comment::create($validated);

    return $this->apiResponse(new CommentResoucre($comment));
}

```

]

2-Located at [ReplyController.php](#).

- **store(StoreReplyRequest \$request)**: Stores a new reply for a comment.

```

public function store(StoreReplyRequest $request)
{
    DB::beginTransaction();

    try {
        $comment = Comment::findOrFail($request->comment_id);

        if ($comment->hasReachedReplyLimit()) {
            return response()->json([
                'error' => 'Reply limit reached',
                'code' => 422,
            ], 422);
        }

        $comment->replies()->create($request->validated());
        $comment->load('replies');

        DB::commit();

        return $this->apiResponse(new CommentResource($comment), 201);
    } catch (\Exception $e) {
        DB::rollBack();
        return response()->json([
            'error' => 'Failed to store reply',
            'code' => 500,
        ], 500);
    }
}

```

Models

- **Comment:** Represents the comments table.
 - **replies():** Defines a one-to-many relationship with the Reply model.
 - **hasReachedReplyLimit():** Checks if the comment has reached the reply limit (5 replies).

```

public function replies(): HasMany
{
    return $this->hasMany(Reply::class);
}

public function hasReachedReplyLimit(): bool
{
    return $this->replies()->count() == 5;
}

```

- **Reply:** Represents the [replies](#) table.
 - **comment():** Defines a many-to-one relationship with the Comment model.

```

public function comment(): BelongsTo
{
    return $this->belongsTo(Comment::class);
}

```

Routes

- **API Routes:** Defined in [api.php](#).

```
Route::prefix('comments')->group(function () {  
    Route::get('/', [CommentController::class, 'index']);  
    Route::get('/{id}', [CommentController::class, 'show']);  
    Route::post('/', [CommentController::class, 'store']);  
    Route::post('/replies', [ReplyController::class, 'store']);  
});
```

- **Web Routes:** Defined in [web.php](#).

```
Route::get('/', function () {  
    return view('comments');  
});
```