Appendix G2

This appendix presents GraphQL queries that replicate the functionality of RESTful API requests for retrieving posts, viewing root comments, fetching replies, and creating nested comments. Each GraphQL query is designed to match its REST request.

```
#1.      Fetching a user's posts:
# mirror RSET GET /api/v1/users/{userId}/posts
query {
  getUser(userId: "a14092e3-cf86-4337-b75e-b76df514385f") {
    posts(page: 0, size: 10) {
      content {
        id
        author {
          id
          name
          avatarUrl
          isGroup
        }
        text
        images {
          id
          uri
          reactions{
           reactionValue
           reactionCount
            participants{
              id
              name
              avatarUrl
              isGroup
          }
        }
      }
     }
    }
   }
 }
}

# 2.     Viewing root comments:
# mirror GET /api/v1/posts/{postId}/comments
query {
  getUser(userId: "a14092e3-cf86-4337-b75e-b76df514385f") {
    post(postId: "6a803ea9-d828-455f-8713-e037fcedf09e") {
      comments(page: 0, size: 10) {
        content {
          id
          postId
          parentCommentId
          author {
            id
            name
            avatarUrl
            isGroup
          }
          text
          images {
            id
            uri
```

```
            }
            reactions {
              reactionValue
              reactionCount
              participants {
                id
                name
                avatarUrl
                isGroup
              }
            }
          }
        }
      }
    }
  }
}
```

#3 Fetching replies to a comment
 # mirror GET /api/v1/posts/{postId}/comments/{commentId}/comments
```
query {
  getUser(userId: "a14092e3-cf86-4337-b75e-b76df514385f") {
    post(postId: "6a803ea9-d828-455f-8713-e037fcedf09e") {
      comment(commentId: "707") {
        id
        postId
        parentCommentId
        author {
          id
          name
          avatarUrl
          isGroup
        }
        text
        images {
          id
          uri
        }
        reactions {
          reactionValue
          reactionCount
          participants {
            id
            name
            avatarUrl
            isGroup
          }
        }
      }
    }
  }
}
```

#4 Writing a nested comment
#mirror POST /api/v1/posts/{postId}/comments/{commentId}/comments
```
mutation {
  createNestedComment(
    postId: "6a803ea9-d828-455f-8713-e037fcedf09e"
    commentId: "707"
    input: { text: "Absolutely, Kadiatou! Jean-Pierre's words beautifully capture the
essence of cultural exchange. It's inspiring to see how sharing traditions and stories
can bring us all closer together.", images: [] }
  ) {
    id
```

```
      postId
      parentCommentId
      author {
         id
         name
         avatarUrl
         isGroup
      }
      text
      images {
         id
         uri
         reactions{
            reactionValue
            reactionCount
            participants{
               id
            }
         }
      }
   }
}
```

Result writing a nested comment

```
{
   "data": {
      "createNestedComment": {
         "id": "12021",
         "postId": "6a803ea9-d828-455f-8713-e037fcedf09e",
         "parentCommentId": "707",
         "author": {
            "id": "99e6700c-593e-497c-8b6a-9f00b53e05ff",
            "name": "Oksana Zarowska",
            "avatarUrl": "/public/api/images/7989b761-0d8a-4546-b8c3-3dc3f0b80088",
            "isGroup": false
         },
         "text": "Absolutely, Kadiatou! Jean-Pierre's words beautifully capture the
essence of cultural exchange. It's inspiring to see how sharing traditions and stories
can bring us all closer together.",
         "images": []
      }
   }
}
```