COM431 Assignment by: **Torok, Adam**
Course Code: **8260** Student ID: **B00798824**
Tables: **Results** and **Terminology**

COMMENT BANK

Tables to be focused on: Results and Terminology

Name: Adam Torok

TABLE OF CONTENTS

COVERAGE.....IV

MODEL.....IV

PHP Artisan Commands (model).....viii

SEEDER.....VIII

The second table which only contains a structure name, abbreviation and id.....xii

PHP Artisan Commands (Seeder).....xiii

ROUTES.....XIII

COMMENTCONTROLLER.....XIV

AJAXCONTROLLER.....XVI

Artisan Commands.....xviii

VIEWS.....XIX

*.app.blade.php.....xix

*.header.blade.php.....xix

*.footer.blade.php.....xix

*.navbar.blade.php.....xx

*.sidebar.blade.php.....1

*.newCommentForm.blade.php.....2

*.scratchpad.blade.php.....5

NPM.....5

webpack.mix.....5

18 .sourceMaps(); myapp.js.....6
mycss.css.....14
npm commands.....18

CRUD – ADD.....19

CRUD – READ.....19

CRUD – UPDATE.....19

CRUD – DELETE.....19

SELECTING COMMENTS.....19

FIGURES.....20

User Page.....20
Filtering.....21
User management.....23
Admin View.....24
Create.....25

EDIT.....27

DELETE COMMENT.....30

USER.....32

CLIPBOARD.....33

COVERAGE

All presented code was written from scratch.

MODEL

Basic CRUD operations called by the built-in static methods (::delete(), ::create()… etc), or extended because the use of table joins.

Comment.php (comment queries)

```php
<?php
namespace App\Models;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Auth;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Comment extends Model
{
    use HasFactory;

    protected $fillable =
['lastName','firstName','comment','tone','email','structureId'];


    //Api Controller redirects here to query
    //the response goes back to Api Controller

    //get all comments
    public function index($structureId=""){
        $data = DB::table('comments')
            ->join('sructures', 'comments.structureId',
'=' ,'sructures.id')
            ->select('comments.*','sructures.code','sructures.name')
            ->orderBy('comments.structureId','asc')
            ->orderBy('comments.id','desc');

        if(Auth::check() == 0){
            $data->where('isApproved',1);
        }

        if($structureId != ""){
```

```
31              $data->where('structureId',$structureId);
32          }
33          $data = $data->get();
34          return $data;
35      }
36
37
38      //show one record
39      public function show($id){
40          $data = DB::table('comments')
41              ->join('sructures', 'comments.structureId',
'=' ,'sructures.id')
42              ->select('comments.*','sructures.code','sructures.name')
43              ->where('comments.id','=',$id)
44              ->get();
45          return $data;
46      }
47
48      //get the last record
49      public static function last(){
50
51          $data = DB::table('comments')
52              ->join('sructures', 'comments.structureId',
'=' ,'sructures.id')
53              ->select('comments.*','sructures.code','sructures.name')
54              ->orderBy('comments.created_at','desc')
55              ->first();
56          return $data;
57      }
58
59      public function deleteComment($id){
60          //delete a single comment
61          $deleted = Comment::find($id);
62          $deleted->delete();
63          return $deleted ? true : false;
64      }
65
66      function toggleApprove($id){
67          //approves a comment Auth check in ajax
68          $oldState = DB::table('comments')->where('id',$id)-
>get(['isApproved']);
69          $newState = $oldState[0]->isApproved == 1 ? 0 : 1;
70          DB::table('comments')->where('id',$id)
71                               ->update(['isApproved' => $newState]);
72
73      }
74  }
```

Sidebar has displaying all the categories with the actual category counts.

Sidebar.php (generating sidebar with the structure counts)

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Auth;

class Sidebar extends Model
{
    use HasFactory;


    //this model returns the structure items for dynamic rendering
    //also, returns with a count of each structure items in the database


    public function index(){
        //return all sidebar items
        $data = DB::table('sructures')->get();
        return $data;
    }

    public function getTopicCountArray(){
        //return as counting all topics we have in the database
        //[2,3,4,5,6,2,0,0,4,5]

        $topicCount = [];

        $approved = Auth::check() == 0 ? "where isApproved=1" : "";
        $count = DB::select("SELECT structureId as id, COUNT(*) as count
FROM laravel.comments $approved group by structureId;");


        return $count;

    }
}
```

PHP Artisan Commands (model)

Php artisan make:migration Sidebar

Php artisan make:migration Comment

Php artisan migrate:fresh

SEEDER

The Comments were assembled in the seeder file.

CommentSeeder.php

```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Str;

class CommentSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {

        $categories = [];

        //[comment,categoryId]

        $terminology = [
            ["Method, Mathematics and Proper Terminology",5],
            ["The method/methodology has not been justified through presentation of potential alternatives and resulting justification of the selected method on a balanced and well reasoned basis.",5],
            ["The method/methodology is not well reported and there is insufficient detail and description; the reader should be able to follow
```

```
the reasoning and development and there should be adequate explanation and
illustration in relation to the approach adopted.",5],
28              ["Important assumptions have not been
addressed/stated.",5],
29              ["Fundamental flaws in methodology have been noted.",5],
30              ["Standard and well accepted approaches have not been
adopted, without clear justification for avoiding these.",5],
31              ["The presentation of mathematics has clear deficiencies
e.g. errors are present; wrong symbols have been used, Greek letters are
absent or incorrect, subscripts and/or superscripts are missing; equation
layout is flawed.",5],
32              ["Important equations/expressions have not been properly
numbered in sequence.",5],
33              ["Key steps are missing in the development of
equations/expressions.",5],
34              ["Excessive detail has been included in the development of
equations/expressions.",5],
35              ["Units or symbols are not consistent with appropriate
conventions e.g. such as the SI system of units.",5],
36              ["Abbreviations and/or acronyms do not follow normal
conventions or have not been explicitly defined at the first mention.",5],
37              ["Results and Analysis",5],
38              ["Assumptions have not been clearly stated.",5],
39              ["There should be some tabular presentation of
results.",5],
40              ["Graphs or graphical presentation have/has not been used
as expected.",5],
41              ["The analysis of the results is too trivial in nature and
a more detailed analysis was necessary (e.g. no confidence limits have
been indicated).",5],
42              ["Too much has been read into the results and sweeping
conclusions have been made, which are not fully justified.",5],
43              ["Statistical analysis aspects are flawed, e.g. they are
incorrect, incomplete or inappropriately applied.",5],
44              ["There is insufficient discussion of results and it has
been wrongly assumed that they speak for themselves.",5],
45              ["Anomalies are present in results which have not been
properly discussed or commented upon.",5],
46              ["Presentation of results involves excessive repetition and
an appendix should have been used for the bulk of the material, once the
first example(s) had been shown.",5],
47              ["Irrelevant material has been included, which does not
serve to enhance the presentation of results and/or related
analysis.",5]];
48
49          $results = [
50              ["Results and Analysis",6],
```

```
51              ["Assumptions have not been clearly stated.",6],
52              ["There should be some tabular presentation of
results.",6],
53              ["Graphs or graphical presentation have/has not been used
as expected.",6],
54              ["The analysis of the results is too trivial in nature and
a more detailed analysis was necessary (e.g. no confidence limits have
been indicated).",6],
55              ["Too much has been read into the results and sweeping
conclusions have been made, which are not fully justified.",6],
56              ["Statistical analysis aspects are flawed, e.g. they are
incorrect, incomplete or inappropriately applied.",6],
57              ["There is insufficient discussion of results and it has
been wrongly assumed that they speak for themselves.",6],
58              ["Anomalies are present in results which have not been
properly discussed or commented upon.",6],
59              ["Presentation of results involves excessive repetition and
an appendix should have been used for the bulk of the material, once the
first example(s) had been shown.",6],
60              ["Irrelevant material has been included, which does not
serve to enhance the presentation of results and/or related
analysis.",6]];
61          //array_push($categories,$spelling);
62          //array_push($categories,$structure);
63          //array_push($categories,$abstracts);
64          //array_push($categories,$introduction_material);
65          array_push($categories,$terminology);
66          array_push($categories,$results);
67          //array_push($categories,$conclusions);
68          //array_push($categories,$references);
69          //array_push($categories,$reflection);
70          //array_push($categories,$plagiarism);
71          //array_push($categories,$appendix);
72
73          //seeding the unapproved comments
74          //if your table is different, just taylor it
75          foreach($categories as $category){
76              for($i = 0; $i < count($category); $i++){
77                  //change the table layout to fir yours
78                  DB::table('comments')->insert([
79                      'comment'       => $category[$i][0],
80                      'firstName'     => "Alan",
81                      'lastName'      => "Webb",
82                      'created_at'    => now(),
83                      'updated_at'    => now(),
84                      'structureId'   => $category[$i][1],
```

```
85                      'isApproved'    => 1,
//0 - not approved - 1 - approved
86                      'email'         => "",
87                      'tone'          => 0,
88                  ]);
89              }
90          }
91
92      } //method
93  } //class
94
```

The second table which only contains a structure name, abbreviation and id

StructureSeeder.php

```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Str;

class StructureSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run() {

        //category names with codes
        $data = [
        ['Structure','ST'],
        ['Abstracts','AB'],
        ['Introduction Material','IN'],
        ['Programming','PR'],
        ['Terminology','TE'],
        ['Results','RS'],
        ['Conclusions','CO'],
        ['Reflection','RF'],
        ['References','RE'],
        ['Appendix','AP'],
        ['Spelling Mistakes','SP'],
        ['Plagilarism','PL']
        ];



        for($i = 0; $i <count($data); $i++){
            DB::table('sructures')->insert([
                'name' => $data[$i][0],
                'code' => $data[$i][1],
                'created_at' => now(),
                'updated_at' => now(),
            ]);
```

```
        }

    }
}
```

PHP Artisan Commands (Seeder)

Php artisan db:seed CommentSeeder

php artisan db:seed StructureSeeder

ROUTES

```php
<?php
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\CommentController;
use App\Http\Controllers\AjaxController;
use Illuminate\Support\Facades\Auth;

Route::get('/', [CommentController::class,'index'])->name('Home');

//CRUD RELATED----------------------------------------
//CREATE
Route::post('/comment', [AjaxController::class,'create']);
//READ 1
Route::get('/comment/{id}',  [AjaxController::class, 'show']);
//READ SOME
Route::get('/comment/cat/{id}',  [AjaxController::class,
'getCommentsByStructureId']);
//READ ALL
Route::get('/comment', [AjaxController::class,'all']);
//UPDATE
Route::patch('/comment/{id}',  [AjaxController::class, 'update']);
//DELETE
Route::delete('/comment/{id}', [AjaxController::class,'delete']);
//CRUD RELATED----------------------------------------
//

Route::get('/lastcomment', [AjaxController::class,'lastComment']);


Route::post('/toggleApprove', [AjaxController::class,'toggleApprove']);

Route::get('/ip', function(){
    //on config change, the docker's mysql ip change
    //call this page to see your ip and set it in the .env
```

```
    //file DB_HOST=x.x.x.x
    dd(Request::ip());
});

Route::get('/contact', function(){
    $data['cards'] = 3;
    return view('home.contact',$data);
});

require __DIR__.'/auth.php';


Route::get('/isAuth',  [CommentController::class, 'isAuth']);

Route::get('/logout', function(){
    Auth::logout();
    return redirect('/');
});

Auth::routes();

Route::get('/home', [App\Http\Controllers\HomeController::class,
'index']);

Route::get('/dashboard', function () {
    return view('welcome');
})->middleware(['auth'])->name('dashboard');
```

COMMENTCONTROLLER

This Controller just renders the page index

CommentController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\Comment;
use App\Models\Sidebar;
use App\Http\Controllers\AjaxController;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class CommentController extends Controller{

    //this is the main controller to display the site
```

```
14
15
16      //rendering all the comments and
17      //building the sidebar
18      public function index(){
19          //all comments
20          //$a = new AjaxController();
21          //$data['comments'] = $a->all();
22
23          //get sidebar elements
24          $sidebar = new Sidebar();
25          $data['sidebar'] = $sidebar->index();
26
27          //get the structure counts
28          $data['sidebarCount']  = $sidebar->getTopicCountArray();
29
30          return view('home.index', $data);
31      }
32
33      function isAuth(){
34          //check whether the user is logged in
35          if(Auth::check()  == 1){
36              return 1;
37          }
38          else{
39              return 0;
40          }
41
42
43      }
44
45  }
```

AJAXCONTROLLER

Ajax Controller returns only JSON data.

AjaxController.php

```php
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Comment;
Use Illuminate\Support\Facades\Auth;

class AjaxController extends Controller {
    //ajax controller always returns json datatype
    //show returns all records
    public function all(){
        $c = new Comment();
        $data = ['isAuth' => Auth::check(), 'comments' => $c->index()];
        return response()->json($data,200);
    }

    //show one record
    public function show($commentId){

        $c = new Comment();
        $result =  $c->show($commentId);
        return response()->json($result);
    }

    //create a record
    public function create(Request $request){
        $c = new Comment();
        $request->validate([
            'lastName' => 'min:3|max:25|required',
            'firstName' => 'min:3|max:25|required',
            'comment' => 'min:3|required',
            'email' => 'min:3|max:25|required',
            'tone' => 'required',
            'structureId' => 'required'
        ],
        ['structureId' => 'Structure']
        );


```

```
40          $c->lastName = $request->input('lastName');
41          $c->comment = $request->input('comment');
42          $c->structureId = $request->input('structureId');
43          $c->tone = $request->input('tone');
44          $c->email = $request->input('email');
45          $c->firstName = $request->input('firstName');
46          $c->created_at = now();
47          $c->updated_at = now();
48          $c->save();
49          $data = $c->all();
50          return response()->json($data);
51      }
52
53      //edit a post
54      public function update(Request $request){
55          $c = new Comment();
56
57          $request->validate([
58              'lastName' => 'min:3|max:25|required',
59              'firstName' => 'min:3|max:25|required',
60              'comment' => 'min:3|required',
61              'email' => 'min:3|max:25|required',
62              'tone' => 'required',
63              'structureId' => 'required'
64          ],
65          ['structureId' => 'Structure']
66          );
67
68          $data = [
69
70              "lastName" => $request->input('lastName'),
71              "comment" => $request->input('comment'),
72              "structureId" => $request->input('structureId'),
73              "tone" => $request->input('tone'),
74              "email" => $request->input('email'),
75              "firstName" => $request->input('firstName'),
76              "updated_at" => now()];
77
78          $c->where('id',$request->input('commentId'))->update($data);
79          $updatedPost = $c->where('id',$request->input('commentId'))-
>get();
80
81          return response()->json($updatedPost,200);
82
83      }
84
85
86
```

```
87      //delete a record
88      public function delete($id){
89          $c = new Comment();
90          $result = $c->deleteComment($id);
91          $code = $result == true ? 200 : 400;
92          $data['message'] = $code == 200 ? "success" : "error";
93          $data = $c->index();
94          return response()->json($data,$code);
95      }
96
97      function lastComment(){
98          $c = new Comment();
99          return response()->json($c::last(),200);
100     }
101
102     function getCommentsByStructureId($id){
103         //returns just the selected catgory of comments
104         $c = new Comment();
105         return response()->json(['isAuth' => Auth::check(), 'comments'
=>$c->index($id)],200);
106     }
107
108     function toggleApprove(Request $request){
109         //approve a comment in the database
110         if(Auth::check() == 0){
111             return response()->json(['error','Not Authorized'],500);
112         }
113         $id = $request->input('id');
114         if($id){
115             $c = new Comment();
116             $c->toggleApprove($id);
117         }
118
119     }
120 }
```

Artisan Commands

Php artisan make:controller AjaxController

Php artisan make:controller CommentController

VIEWS

*.app.blade.php

```
<!DOCTYPE html>
<html lang="en">
@include('home.components.header')
<body>
@include('home.components.navbar')
      @yield('content')
@include('home.components.footer')
</html>
```

*.header.blade.php

```
  <head>
  <title>@yield('title')</title>
  <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
  <meta name="description" content="HTML5 skeleton index.html">
  <meta name="keywords"
content="html5,skeleton,index,homepage,jquery,bootstrap">
  <meta name="author" content="Arul John">
  <link href="{{ asset('css/app.css') }}" rel="stylesheet">
  <link href="{{ asset('css/mycss.css') }}" rel="stylesheet">
  <meta name="csrf-token" content="{{ csrf_token() }}">
  <script>

    window.AuthUser = "{{{ (Auth::user()) ? Auth::user() : null }}}";

  </script>
  </head>

```

*.footer.blade.php

```

<!-- Latest compiled and minified JavaScript -->
<script src="{{ asset('js/app.js') }}" defer></script>
<script src="{{ asset('js/myapp.js') }}" defer></script>
</body>
```

*.navbar.blade.php

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark shadow fixed">
    <a class="navbar-brand" href="/"><?php echo Auth::check() == 1 ? "admin@" : "user@";?>@yield('title')</a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExample05" aria-controls="navbarsExample05" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>

      <div class="collapse navbar-collapse" id="navbarsExample05">
        <ul class="navbar-nav mr-auto">
          <li class="nav-item active">
            <a class="nav-link" href="#">All <span class="sr-only">(current)</span></a>
          </li>
          <li class="nav-item">
            <a class="nav-link" data-toogle="modal" data-target="#newComment" href="#">Add a Comment</a>
          </li>
          @if(Auth::check())
          <li class="nav-item">
              <a class="nav-link" href="/logout">Logout</a>
          </li>
          @else
          <li class="nav-item">
            <a class="nav-link" href="/login">Login</a>
          </li>
          @endif
          <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="http://example.com" id="dropdown05" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">Dropdown</a>
            <div class="dropdown-menu" aria-labelledby="dropdown05">
              <a class="dropdown-item" href="#">Action</a>
              <a class="dropdown-item" href="#">Another action</a>
              <a class="dropdown-item" href="#">Something else here</a>
            </div>
          </li>
        </ul>
      </div>
    </nav>
```

COM431 Assignment by: **Torok, Adam**
Course Code: **8260** Student ID: **B00798824**
Tables: **Results** and **Terminology**

*.sidebar.blade.php

```
<div class="sidebar">
    <!--Rendering the page sidebar -->

    <!--if the sidebar has element, iterating them -->
        @if($sidebar->isNotEmpty() == true)
            <button type="button" id="openNewCommentModal"
                    class="shadow newForm btn btn-primary btn-sm btn-block"
                    data-toggle="modal" data-target="#newComment">
                    Add a new comment</button>
        <div class="list-group shadow">
          <a href="#" class="filterButton list-group-item list-group-item-action"
             aria-current="true" data-structureId="">
            <div class="d-flex w-100 justify-content-between">All</div>
          </a>

        </div>
        @for($i=0;$i<count($sidebar);$i++)
        @php($c = 0)

        <div class="list-group shadow">
            <!--Category buttons as filter -->
          <a href="#" class="filterButton list-group-item list-group-item-action"
             aria-current="true" data-structureId="{{$sidebar[$i]->id}}">
            <div class="d-flex w-100 justify-content-between"> {{ $sidebar[$i]->name}}


                <!--Category Counts -->
                <span id="counter_{{$sidebar[$i]->code }}"
                      class="badge bg-primary rounded-pill topicbadge">
                    <!--render the counter -->
                        @for($j=0;$j<count($sidebarCount);$j++)
                                    @if($sidebarCount[$j]->id == $sidebar[$i]->id)

                                        @php( $c = $sidebarCount[$j]->count )
                                        @break
                                    @endif
                        @endfor
                        {{$c}}
                </span>
```

```
            </div>
          </a>
        </div> <!--end listgroup -->
        @endfor
            <button type="button" id="openscratchPad"
                    class="shadow newForm btn btn-primary btn-sm btn-block"
                    data-toggle="modal" data-target="#scratchPad">
                    View Clipboard</button>
        @else   <!--sidebar has no element -->
            <h1 class="stcructureTitle">No Cat's</h1>
            <p>Please run:
            <quote>artisan db:seed StructureSeeder</quote>
            </p>
    @endif
</div>
```

*.newCommentForm.blade.php

```
<!--New Comment Modal -->
<div class="modal fade" id="newComment" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">Add a new comment</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                <div id="errormsg"></div>
                <form class="form-horizontal" id="newCommentForm">
                    @csrf
                    <input type="hidden" id="commentId" value="">
                    <div class="row">
                        <div class="col">
                            <div class="col-xs-8"> <input id="firstName" name="firstName"  placeholder="Contributor's First Name" type="text" class="form-control" required="required"> </div>
                        </div>
                        <div class="col">
```

```
21                          <div class="col-xs-8"> <input id="lastName"
placeholder="Contributor's Last Name" name="lastName" type="text" class="form-
control" required="required"> </div>
22                      </div>
23                  </div>
24
25                  <div class="form-group">
26                      <div class="col-xs-8">
27                          <textarea id="comment"  placeholder="Description"
name="comment" cols="35" rows="5" class="form-control"
required="required"></textarea>
28                      </div>
29                  </div>
30                  <div class="form-group">
31                      <div class="row">
32                          <div class="col">
33
34                              <select id="structureId" name="structureId"
class="custom-select form-control">
35                                  <option value="">Please select</option>
36                                  @for($i=0;$i<count($sidebar);$i++)
37                                      <option value="{{ $sidebar[$i]->id }}">{{
$sidebar[$i]->name }}</option>
38                                  @endfor
39                              </select>
40                          </div>
41                          <div class="col">
42                              <input id="email" name="email" placeholder="e-
mail" type="text" class="form-control" required="required">
43                          </div>
44                      </div>
45                      <select id="tone" name="tone" class=" toneSelector
custom-select form-control">
46                          <option value="">Please select the tone of the
comment</option>
47                          <option value="1">Positive</option>
48                          <option value="0">Negative</option>
49                      </select>
50                  </div>
51              </form>
52          </div>
53          <div class="modal-footer">
54              <button type="button" class="closeNewForm btn btn-danger"  data-
dismiss="modal">Close</button>
```

```
55                 <button type="button" class="newCommentFormSaveButton NewComment
btn btn-primary">Save changes</button>
56                 <button type="button" class="newCommentFormEditButton NewComment
btn btn-primary hidden">Edit</button>
57             </div>  <!--modal footer -->
58         </div>      <!--modal body -->
59     </div>
60 </div>
61 </div>
62 </div>
63
64 <!-- Confirm Modal -->
65 <div class="modal fade" id="confirmModal" tabindex="-1" role="dialog" aria-
labelledby="confirmedModal" aria-hidden="true">
66   <div class="modal-dialog modal-dialog-centered" role="document">
67     <div class="modal-content">
68       <div class="modal-header">
69         <h5 class="modal-title" id="exampleModalLongTitle">Comment has Been
Added</h5>
70         <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
71           <span aria-hidden="true">&times;</span>
72         </button>
73       </div>
74       <div class="modal-body">
75           <p>Author: <span id="confirmedAuthor"></span></p>
76           <p>Structure: <span id="confirmedStructure"></span></p>
77           <p>Comment: <span id="confirmedComment"></span></p>
78           <p>Confirmed e-mail: <span id="confirmedEmail"></span></p>
79           <p>Tone: <span id="confirmedTone"></span></p>
80       </div>
81       <div class="modal-footer">
82         <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
83       </div>
84     </div>
85   </div>
86 </div>
```

*.scratchpad.blade.php

```
<div class="modal fade" id="scratchpad" tabindex="-1" role="dialog" aria-labelledby="scratchpad" aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">Clipboard</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                <textarea id="scratchpadArea" cols="30" rows="10"></textarea>
                <p>Selecting less or more comments will overwrite this area</p>
            </div>
            <div class="modal-footer">
                <button type="button" id="toClipboard" class="btn btn-primary">Copy To Clipboard</button>
            </div>
        </div>
    </div>
</div>
```

NPM

webpack.mix

```
const mix = require('laravel-mix');

/*
 |--------------------------------------------------------------------------
 | Mix Asset Management
 |--------------------------------------------------------------------------
 |
 | Mix provides a clean, fluent API for defining some Webpack build steps
 | for your Laravel application. By default, we are compiling the Sass
 | file for the application as well as bundling up all the JS files.
 |
 */

mix.js('resources/js/app.js', 'public/js')
```

```
.js('resources/js/myapp.js', 'public/js')
    .css('resources/css/mycss.css.scss', 'public/css')
    .sass('resources/sass/app.scss', 'public/css')
    .sourceMaps();
```

myapp.js

```
$.ajaxSetup({
    headers: {
        'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
    }});

let currentscrollpos;

function renderAllComments(structure=null){
    console.log("Rendering all comments");
    //render all comments
    $('#tbody').empty();
    const url = structure == null || structure == "" ? "/comment" :
"comment/cat/" + structure
    $.ajax({
        type : "GET",
        url : url,
        dataType : "json",
        success: function(data){
            var output = "";
            for(let i=0;i < data.comments.length;i++){

                //action button itional options
                var e = data.comments[i];
                var tone = e.tone == 1 ? "Positive" : "Negative";
                var isApproved = e.isApproved == 1 ? "Approved" : "Pending";
                var isApprovedClass;
                var edit;

                if(data.isAuth == true){
                    isApprovedClass = e.isApproved == 1 ? "approved" : "";
                    edit = `<div class="btn-group" role="group" aria-
label="Basic example">
                        <button data-rowid="${e.id}"
                        data-action="approve" type="button" class="actionButton
                        btn approveComment btn-sm btn-success">A</button>

                        <button data-rowid="${e.id}" data-action="edit"
```

```
 36                         type="button" class="actionButton btn  btn-sm btn-
primary">E</button>
 37
 38                         <button data-rowid="${e.id}" data-action="delete"
type="button"
 39                         data-struct="${e.code}" class="actionButton btn btn-sm
 40                         btn-danger">D</button>
 41                         </div>`;
 42                 }
 43
 44                 output +=
 45                     `<tr id="comment_${e.id}" class="datarow ${
{isApprovedClass}">
 46                     <td><div class="form-check">
 47                       <input class="form-check-input clipboardBox"
type="checkbox" value="${e.id}" />
 48                     </div></td>
 49                     <td class="bold">${e.code}${e.id}</td>
 50                     <td>${e.firstName} ${e.lastName}</td>
 51                     <td id="commentText_${e.id}">${e.comment}</td>
 52                     <td>${tone}</td>
 53                     <td>${isApproved}</td>`;
 54                 output += data.isAuth == true ?  `<td>${edit}</td>` : "";
 55                 output += `</tr>`;
 56             }
 57             $('#tbody').empty();
 58             $('#tbody').append(output);
 59             bindActionButtons();
 60         } //success
 61     });   //ajax
 62 }
 63
 64 renderAllComments();
 65
 66 // update or save a new comment
 67 $('.newCommentFormSaveButton').on('click',function(){
 68
 69     //getting the form elements
 70     var data = {
 71         'firstName' : $('#firstName').val(),
 72         'lastName' : $('#lastName').val(),
 73         'email' : $('#email').val(),
 74         'tone' : $('#tone').val(),
 75         'comment' : $('#comment').val(),
```

```
        'structureId' : $('#structureId').val(),
        'commentId' : $('#commentId').val(),
    }

    console.log("Save button hit with commentId: " + data.commentId);
    var type = data.commentId.length > 1 ? "PATCH" : "POST";
    var url = data.commentId.length > 1 ? "/comment/" + data.commentId :
"/comment";
    console.log("Request type is: " + type);

    //if the hidden field not empty, that means we do a UPDATE request
    $.ajax({
        type : type,
        url : url,
        dataType : "json",
        data : data,
        success: function(data){
            $('#newComment').modal('hide');
            $('#newCommentForm').trigger('reset');
            output = "";
            var tone = data.tone == 1 ? "Positive" : "Negative";
            var isApproved = data.isApproved == 1 ? "Approved" : "Pending";
            var approve_disabled = isApproved == 1 ? "disabled" : "";
            var edit_disabled = isApproved == 1 ? "disabled" : "";
            console.log(data[0].id);

            var edit  = `<div class="btn-group" role="group">

              <button data-rowid="${data.id}" ${approve_disabled}
              data-action="approve" type="button" class=" {{$isApprovedClass}}
              actionButton btn btn-sm btn-success">A</button>

              <button data-rowid="${data.id}" ${edit_disabled}
              data-action="edit" type="button" class="actionButton
              btn  btn-sm btn-primary">E</button>

              <button data-rowid="${data.id}" data-action="delete"
              data-struct="${data.code}" type="button" class="deleteButton
              actionButton btn btn-sm btn-danger">D</button>
              </div>`;

            output += `<tr class="new" id="${data.id}">
            <td>
            <div class="form-check">
```

```
119             <input class="form-check-input clipboardBox"
120                 type="checkbox" value="${data.id}" />
121             </div>
122             </td>
123             <td class="bold">${data.code}${data.id}</td>
124             <td>${data.firstName} ${data.lastName}</td>
125             <td>${data.comment}</td>
126             <td>${tone}</td>
127             <td>${isApproved}</td>
128             <td>${edit}</td>
129             </tr>`;
130             //bind buttons
131             bindActionButtons();
132             //a modal to displaying the values
133             showConfirmedModal();
134             //render all comments
135             renderAllComments();
136             $("html, body").animate({ scrollTop: window.currentscrollpos },
500);
137         },
138 
139         error: function(data){
140             //rendering errors above the form
141             var errors = Object.entries(data.responseJSON.errors).map(
142                 ([key, value]) => ({key,value})
143             );
144             errorOutput = "";
145             for(let i=0;i < errors.length ;i++){
146                 errorOutput += `<div class="alert alert-danger" role="alert">
147                                     ${err = errors[i].value[0]}
148                                 </div>`;
149             }
150             $('#errormsg').empty().html(errorOutput);
151         },
152 
153     });
154 });
155 
156 
157 function bindActionButtons(){
158     //bind action button records
159     //the problem is here the DOM loads faster than the ajax query
160     //so have to call this after the buttons are loaded
161 
```

```
162     $('closeNewForm').on('click', function(){
163         $('#newCommentForm').trigger("reset");
164         $('#errormsg').empty();
165     });
166 
167     $('.newForm').on('click', function(){
168         $('#newCommentForm').trigger("reset");
169         $('#errormsg').empty();
170     });
171 
172     $('.actionButton').on('click', function(){
173 
174         //get the action of the button and the id
175         var rowId = this.getAttribute('data-rowid');
176         var action = this.getAttribute('data-action');
177         window.currentscrollpos = $(window).scrollTop();
178 
179         switch(action){
180                 //if the action is
181             case "edit":
182                 console.log('edit');
183                 console.log('query on id: ' + rowId);
184 
185                 //request a full record and fill the data back
186                 $.ajax({
187                     type : "GET",
188                     url : "/comment/" + rowId,
189                     dataType : "json",
190                     success: function(data){
191                         //fill back the form
192                         $('#firstName').val(data[0].firstName);
193                         $('#lastName').val(data[0].lastName);
194                         $('#comment').val(data[0].comment);
195                         $('#email').val(data[0].email);
196                         $('#structureId').val(data[0].structureId);
197                         $('#tone').val(data[0].tone);
198                         $('#commentId').val(data[0].id);
199                         $('#newComment').modal('show');
200                         console.log(data);
201                     }
202                 });
203 
204                 //fill up modal
205                 break;
```

```
206             case "delete":
207                 //ask if not approved or permission
208                 //modal yes - or - no
209                 $('#deleteModal').modal('show');
210                 $('#deleteCommentId').attr('data-id',rowId);
211                 break;
212             case "approve":
213                 console.log('approve');
214                 const commentId = this.getAttribute("data-rowid");
215                 console.log("What?: " +commentId);
216                 $.ajax({
217                     type : "POST",
218                     url : "/toggleApprove",
219                     dataType : "json",
220                     data : {
221                         id : commentId,
222                     }});
223                 renderAllComments();
224                 break;
225         }
226     });
227 
228     $('.form-check-input').on('change',function(){
229         //if we check a checkbox, it push the comment
230         clipboard = "";
231 
232         var checkboxes = document.getElementsByClassName('form-check-input');
233 
234         for(let i=0;i < checkboxes.length ;i++){
235             var td = "";
236 
237             if(checkboxes[i].checked){
238                 var selector = '#commentText_'+checkboxes[i].value;
239                 td = $(selector).text();
240                 clipboard += td + "\n---------------\n";
241             }
242 
243         }
244         $('#scratchpadArea').text(clipboard);
245         $('#scratchpadArea').val(clipboard);
246         console.log(clipboard);
247     });
248 
249 
```

```
250 }
251
252 //delete a comment (click delete in modal)
253 $('#deleteCommentId').on('click',function(){
254     //
255     var commentId = this.getAttribute('data-id');
256     var structureCode = this.getAttribute('data-struct');
257     $.ajax({
258         type : "DELETE",
259         url : "/comment/" + commentId,
260         dataType : "json",
261         success : function(data){
262             $('#counter_'+structureCode).val(
263                 $('#counter_'+structureCode).val() + 1
264             );
265         },
266     }).then( function(){
267         $('#tbody').empty();
268         renderAllComments();
269     });
270
271
272
273     //if the delete is cancelled, sets the comment's id to ""
274     $('#deleteModal').modal('hide');
275     var commentId = this.setAttribute('data-id', "");
276 });
277
278 //click on the clipboard button will open the
279 //clipbaord modal and show the selection
280 $('#openscratchPad').on('click', function(){
281     $('#scratchpad').modal('show');
282 });
283
284 //on close the new comment form modal
285 //it will empty the form and hide the modal
286 $('.closeNewForm').on('click', function(){
287     $('#newCommentForm').trigger("reset");
288     $('#errormsg').empty();
289 });
290
291 function showConfirmedModal(){
292     //show the confirmed data modal, display with the last comment data
293     $.ajax({
```

```
294         type : "GET",
295         url : "/lastcomment",
296         dataType : "json",
297         success: function (data){
298 
299             let confirmedTone = data.tone == 1 ? "Positive" : "Negative";
300             $('#confirmedAuthor').text(data.firstName + " " + data.lastName);
301             $('#confirmedComment').text(data.comment);
302             $('#confirmedTone').text(confirmedTone);
303             $('#confirmedStructure').text(data.name);
304             $('#confirmedEmail').text(data.email);
305             $('#confirmModal').modal('show');
306         }});
307 }
308 
309 function isAuth(){
310     var result = $.ajax({
311         type : "GET",
312         url : "/isAuth",
313         dataType : "text",
314         success: function (data){
315             if(data.reponseText == 1){
316                 console.log("Ajax isAuth returned: " + data);
317                 return true;
318             }
319         }});
320     return result;
321 }
322 
323 
324 $('.filterButton').on('click', function(){
325     const structureId = this.getAttribute('data-structureId');
326     console.log("What?: " + structureId);
327     renderAllComments(structureId);
328     return false;
329 })
330 
331 
332 $('#toClipboard').on('click', function(){
333 
334             $("#scratchpadArea").select();
335                 document.execCommand('copy');
336 
337 });
```

mycss.css

```
body{
    background: url('https://1.bp.blogspot.com/-HGLIXpE9UsQ/Uh94RRadE7I/AAAAAAAAHVI/tU9EMm-8vCA/s640/Seamless-Background-Paper4.jpg');
    font-family: Segoe UI,Helvetica Neue,Helvetica,Lucida Grande,Arial,Ubuntu,Cantarell,Fira Sans,sans-serif;
    font-size: 12px;
}

.stcructureTitle{
    text-align: center;
}


nav{
    margin-bottom: 40px;
    width: 100%;
}

.list-group{
    min-width: 200px;
}

.sidebar{
    position: fixed;
    top: 100px;
}

.navbar{
    position:fixed;
    z-index: 1;
}

h1:first-child{
    margin-top: 90px;
    text-align: center;
}


#openscratchPad{
    margin-top: 10px;
}
```

```
40
41 #scratchpadArea{
42     width: 100%;
43     min-height: 400px;
44 }
45
46 .form-check input{
47     margin: auto;
48     position: relative;
49     top: 3px;
50 }
51 .new{
52     background:#66ffff;
53     font-style:italic;
54 }
55
56 #commentTable  tr  td{
57     vertical-align: middle;
58 }
59
60 .bold{
61     font-weight:bold;
62 }
63
64
65 .aOffset{
66     margin-top: -100px;
67     padding-top: 100px;
68 }
69
70
71 .topicbadge{
72     border-radius: 50%;
73     color: white;
74     line-height: 20px;
75     width: 20px;
76 }
77
78 .approved, .approved:hover{
79     background: #00cc44 !important;
80     border-color: #00cc44;
81 }
82
83 .toneBadge{
```

```
    float: right;
    line-height: 12px;
    margin-left: 10px;
}

.newForm{
    margin-bottom: 10px;
}

.hidden{
    display: none;
}

.form-control{
    font-size: 12px;
}

#tone{
    margin-top:15px;
}

.note-editor{
    margin-top: 10px;
}

textarea{
    margin-top: 15px;
    min-height: 70px;
}

.toneLabel{
    margin-right: 20px;
}

.submitNewComment{
    margin:auto;
}

.submittedBy{
    color: white;
    margin-left: 30px;
}

.authorLink{
```

```
    color: white;
    font-weight: bold;
}

.commentId{
    font-weight: bold;
    margin-left: 10px;
    margin-right: 10px;
}

.form-check{
    display: inline;
}

.authorLink:hover{
    color: #ccc;
    text-decoration: none;
}

.card-header{
    padding: 3px;
}

.card{
    margin-bottom: 100px;
}

.card-body{
    padding: 0;
}

.copyBtn{
    padding: 1px 6px;
}

.shadow{
    -moz-box-shadow: 8px 10px 7px -7px rgba(0,0,0,0.43);
    -webkit-box-shadow: 8px 10px 7px -7px rgba(0,0,0,0.43);
    box-shadow: 8px 10px 7px -7px rgba(0,0,0,0.43);
}

#select{
    font-size: 12px;
    height:25px;
```

```
172     padding-left: 10px;
173     padding: 0px;
174 }
175 
176 .alert{
177     margin-bottom: 10px;
178     padding: 0.25rem 1.25rem;
179 }
```

npm commands

Npm install

npm run dev

npm run watch

CRUD – ADD

- AjaxController.php above
- Model/Comment.php above
- myapp.js above

CRUD – READ

- AjaxController.php above
- Model/Comment.php above
- myapp.js above above

CRUD – UPDATE

- AjaxController.php above
- Model/Comment.php above
- myapp.js above

CRUD – DELETE

- AjaxController.php above
- Model/Comment.php above
- myapp.js above

SELECTING COMMENTS

Myaspp.js above above

FIGURES

User Page

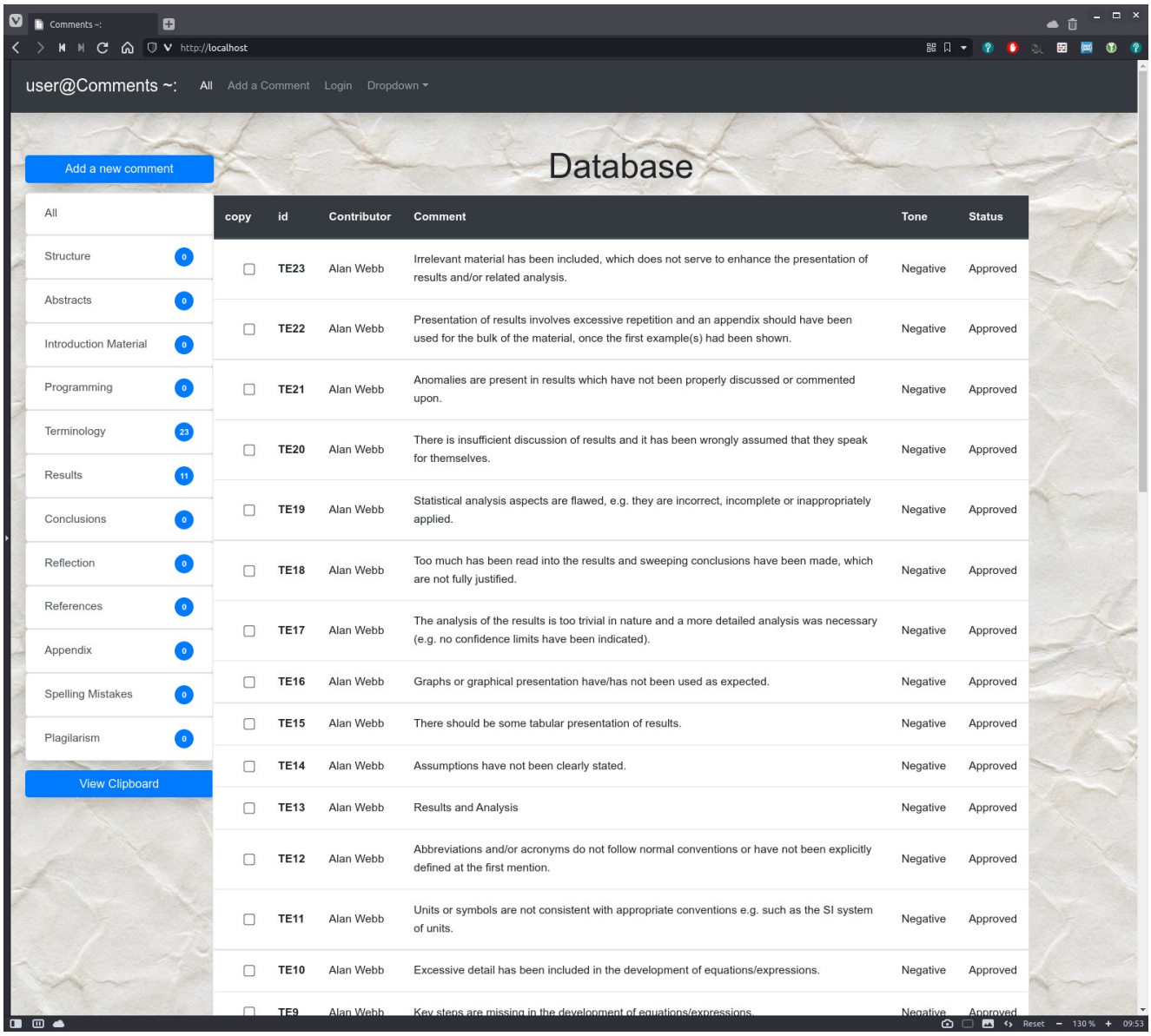


Figure 1: Users Page

The application is one page application. Users and admins arrives at the same page. Every logged in user counts as an admin. Users have no highlighted comments, because only the approved comments are displayed. The login state can be determined by the top left corner. ‘user@comments’ rendered to the user and ‘admin@comments’ rendered to the admin users. Action buttons are hidden.

Filtering

Every User can filter the comments, clicking on the sidebar.

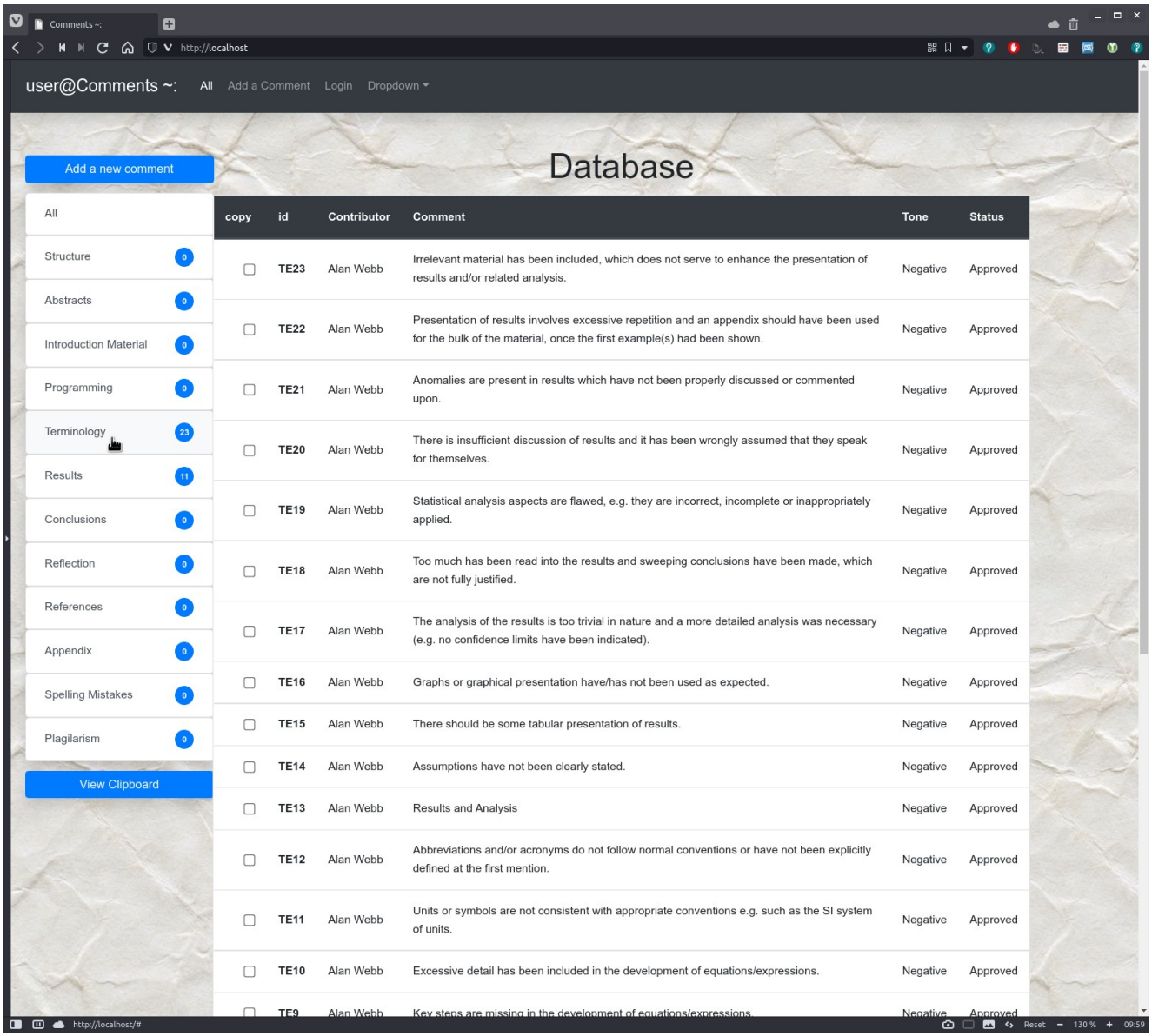


Figure 2: Filtering

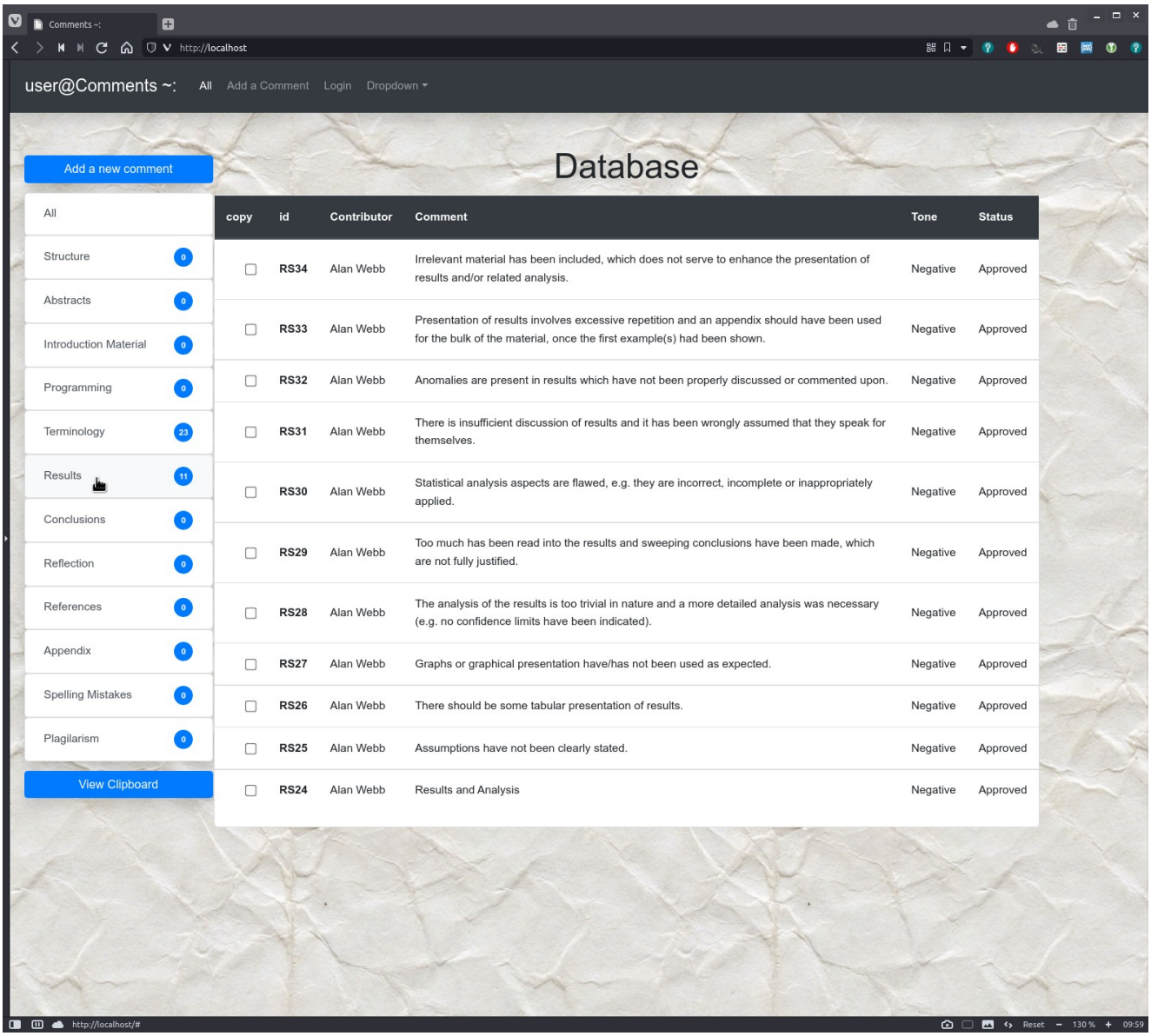


Figure 3: Filtering

User management

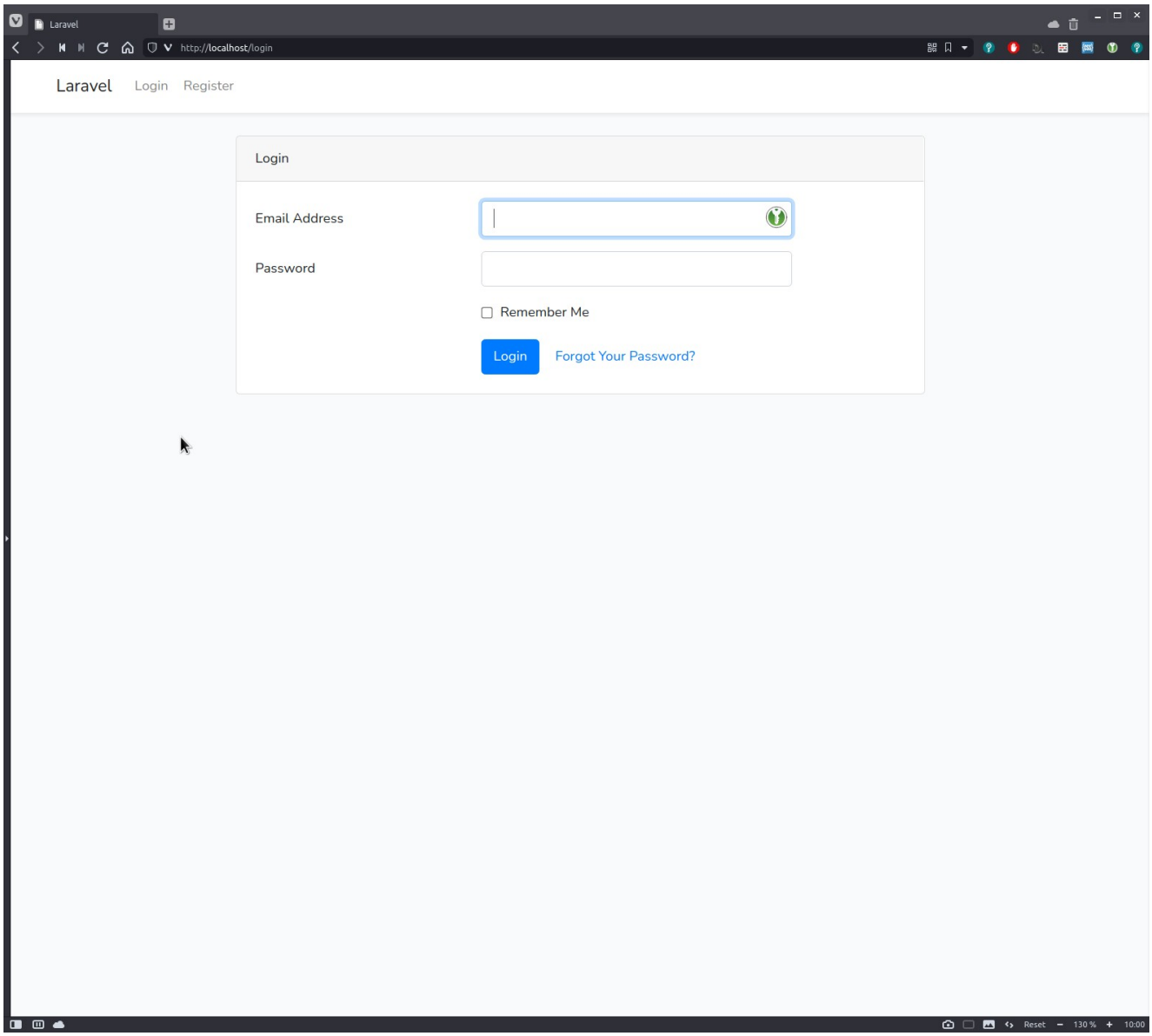
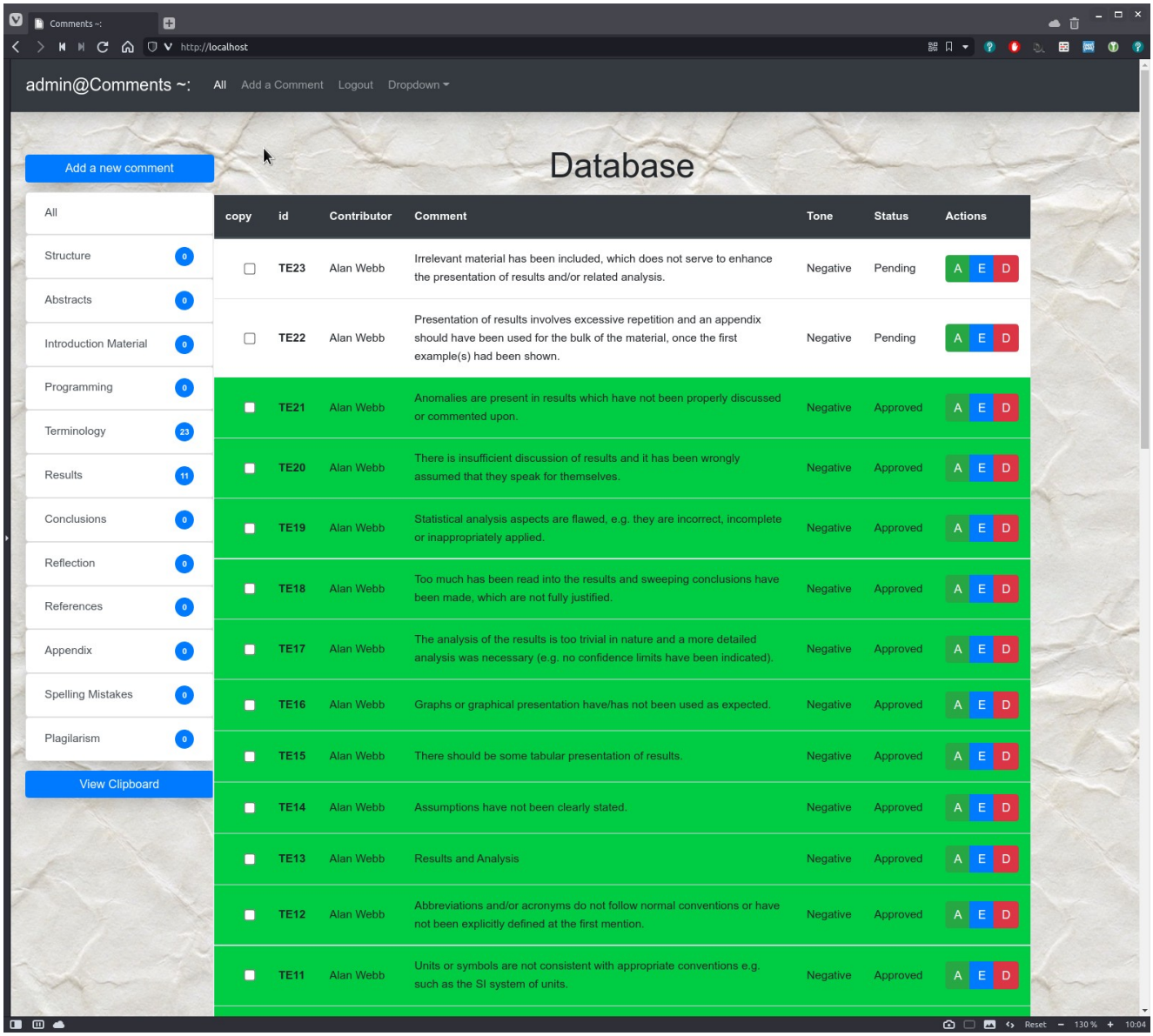


Figure 4: Login

Admin View

Admins can see every comment. The Approved comments are highlighted in green. Three action buttons provided ‘A’,’E’,’D’ for ‘Approve’, ‘Edit’,’Delete’. Pending comments rendered as plain white. Approve uses a toggle mechanism with visual confirmation (table row background change).

Figure 5: Admin view



Create

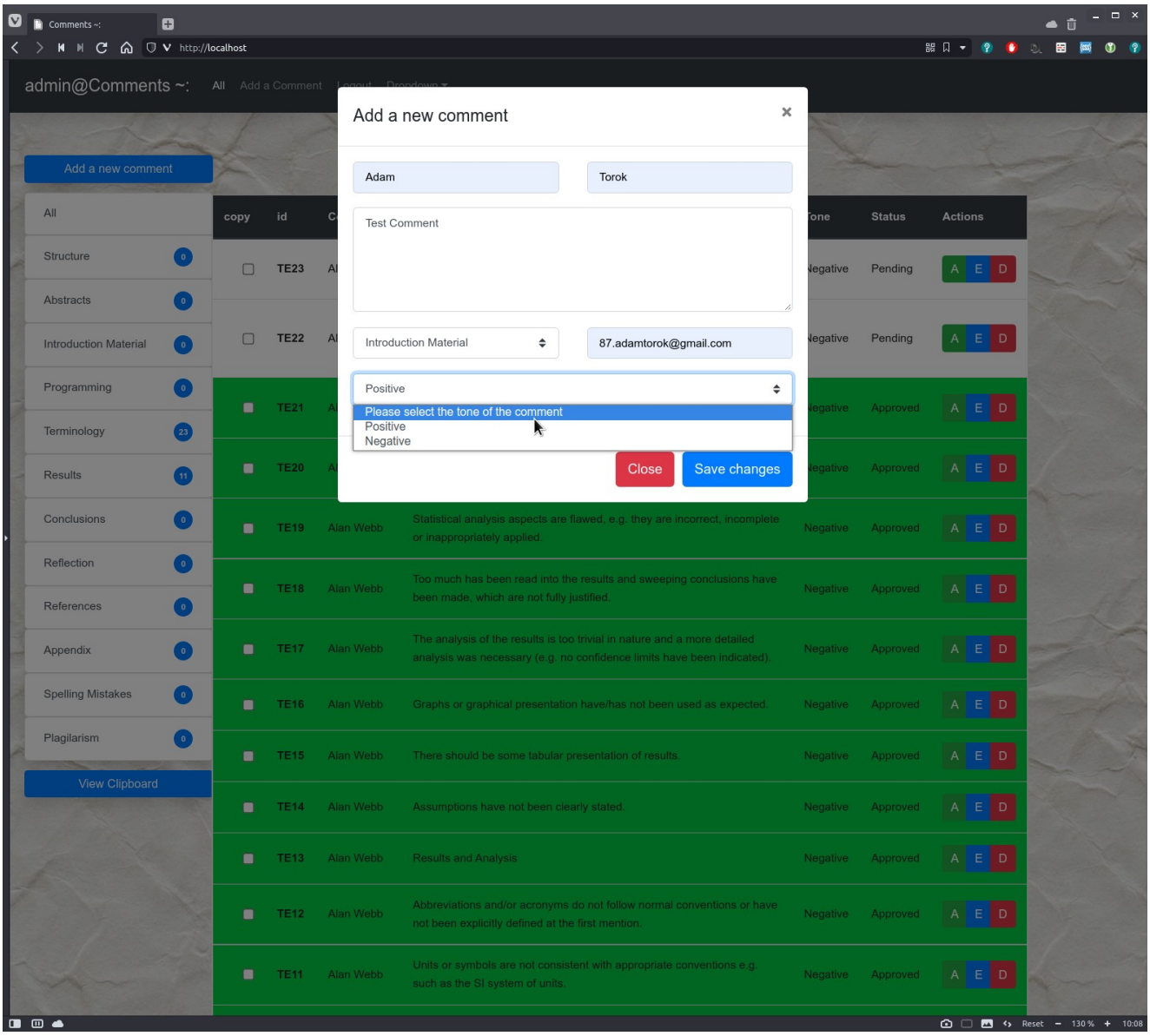


Figure 6: Create comment

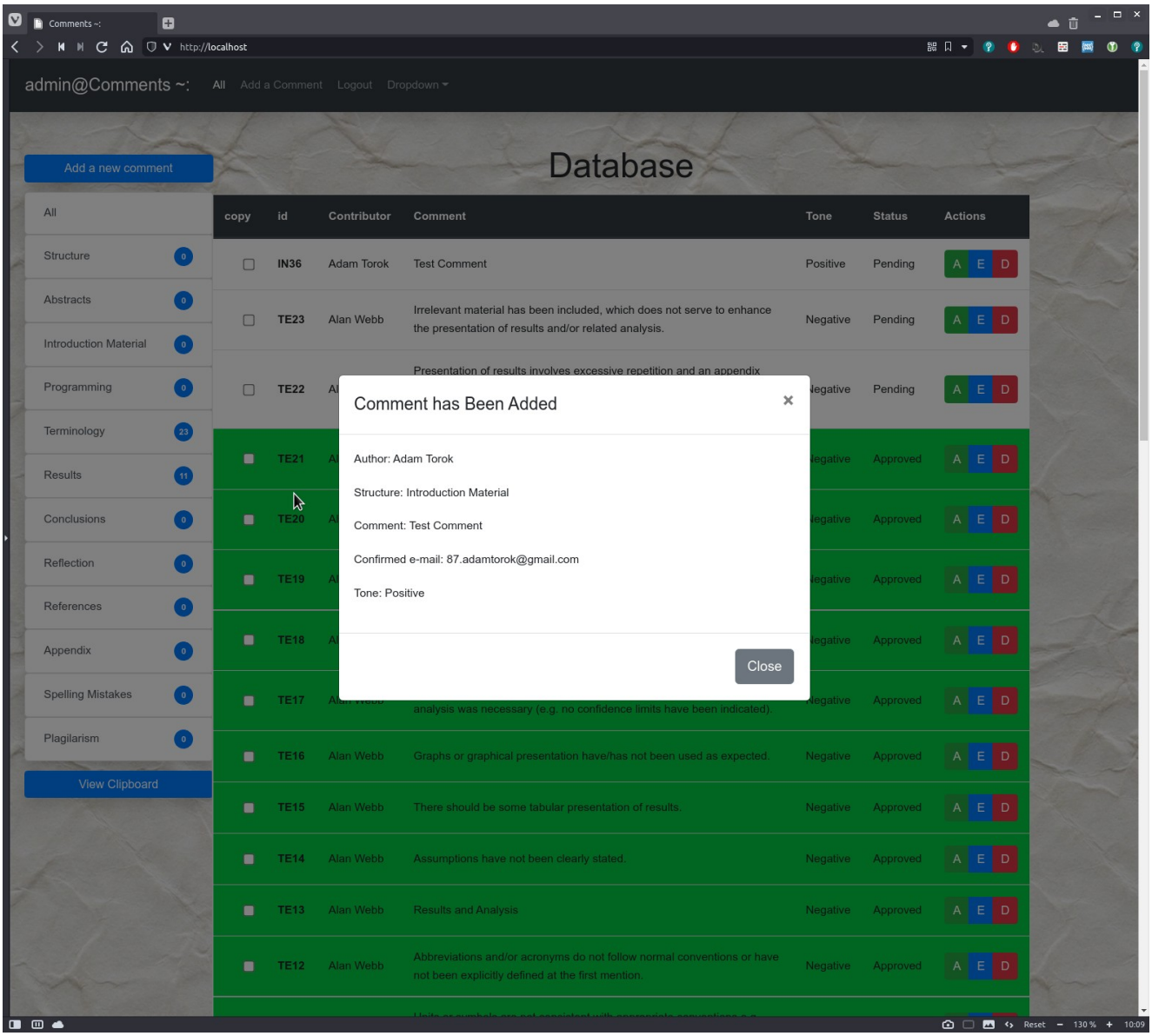
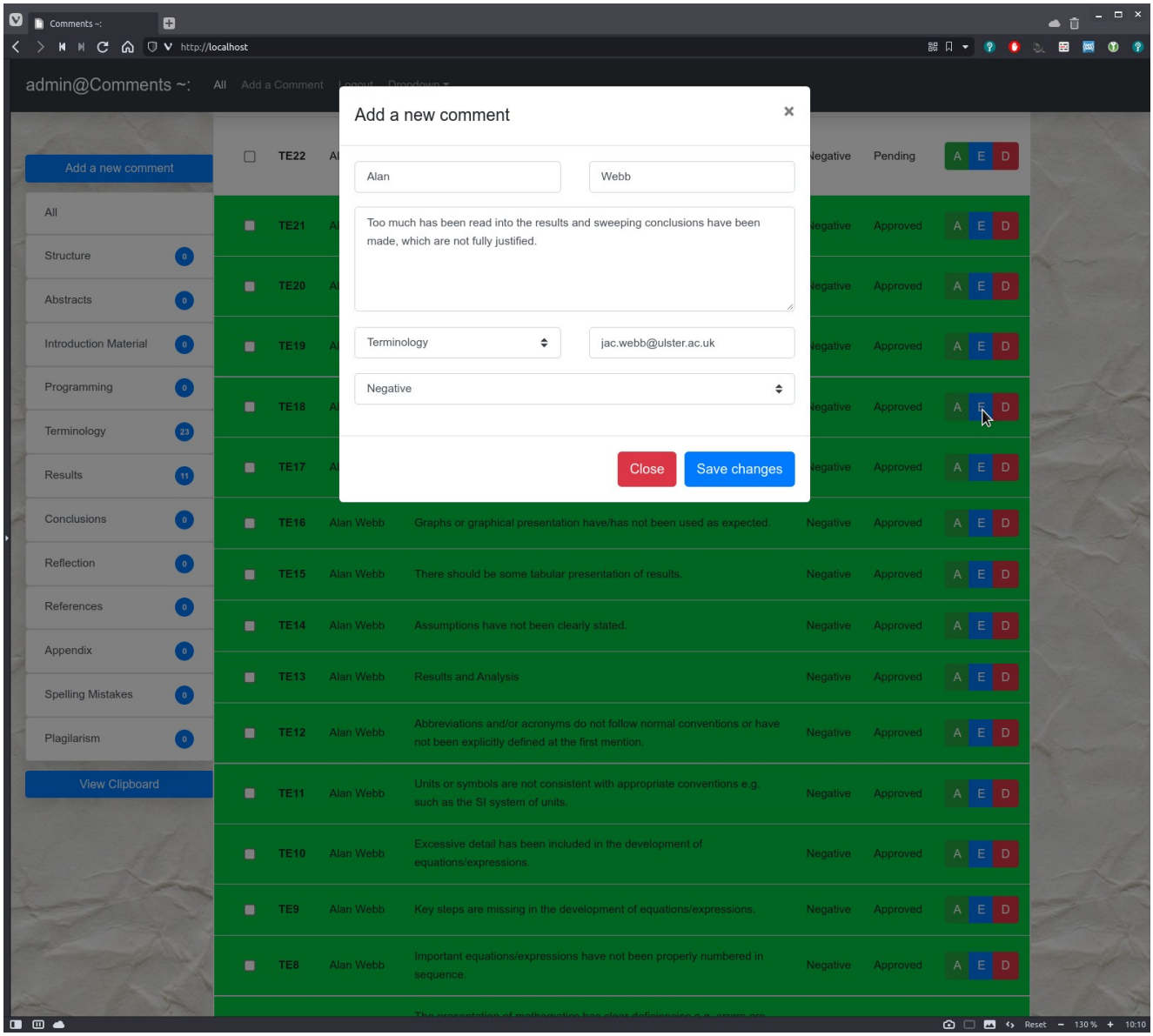


Figure 7: Confirmation

EDIT

On Edit the form is populated with the old values.



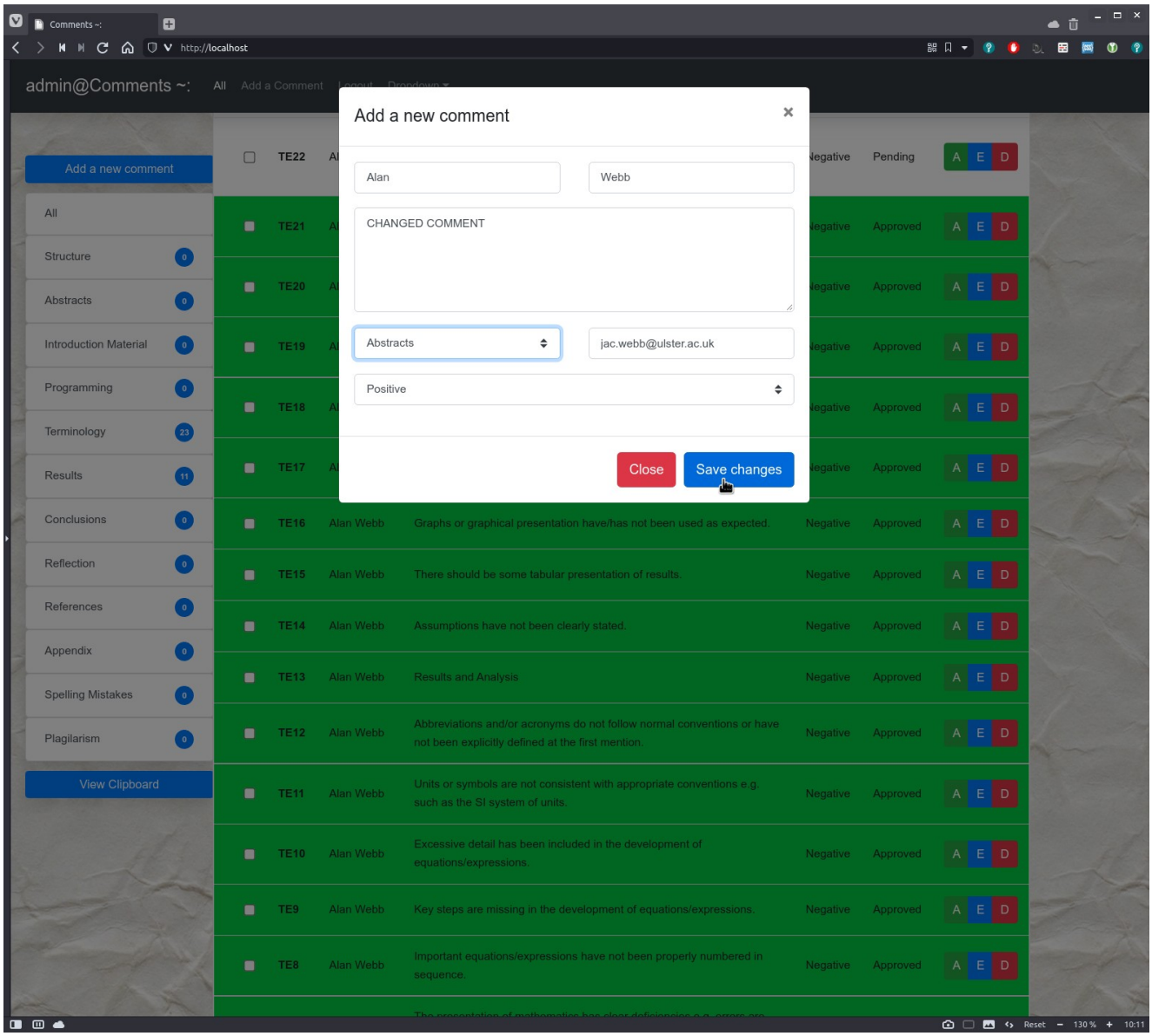


Figure 8: Change the comment

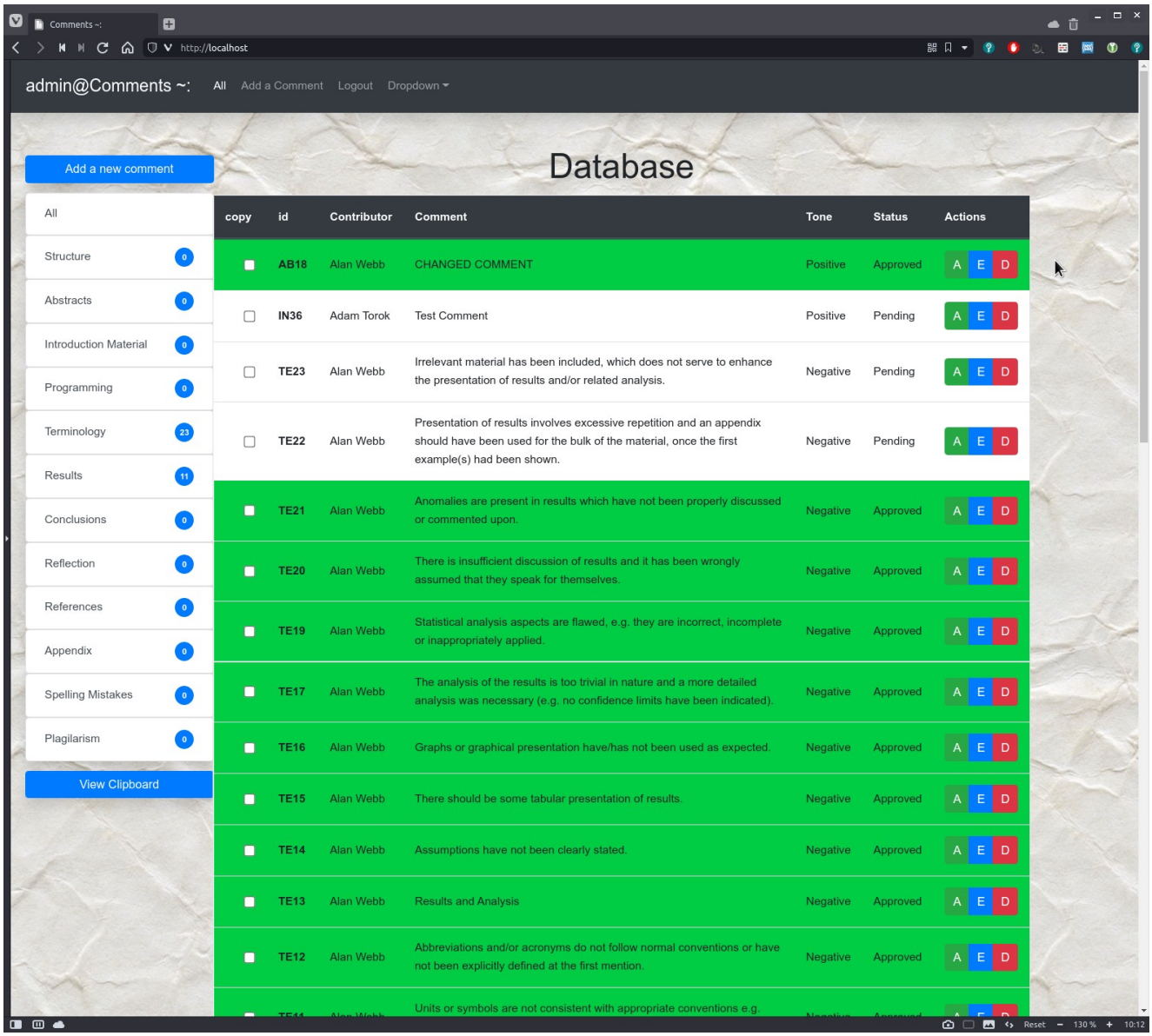
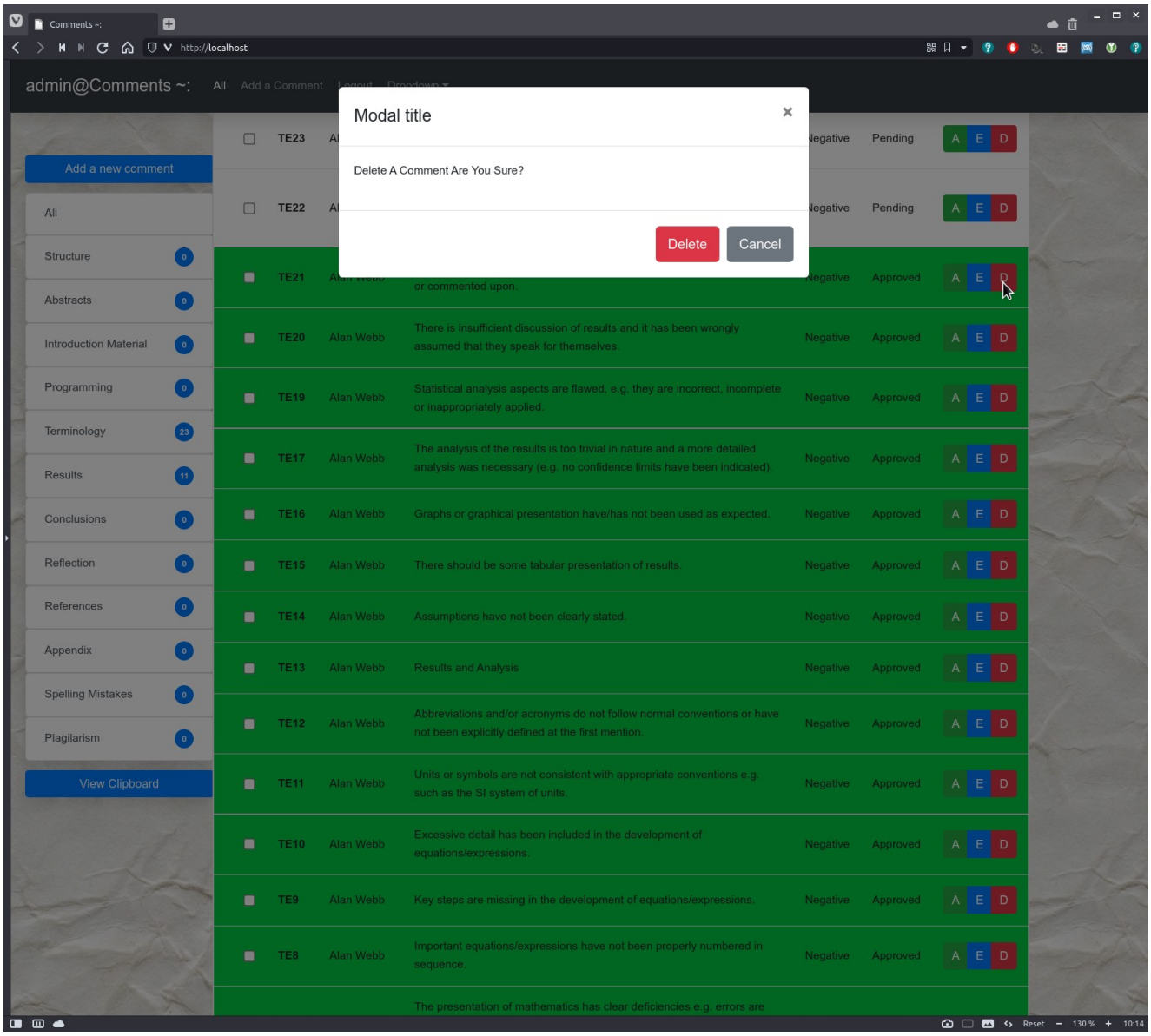


Figure 9: Comment Has Been Changed

DELETE COMMENT



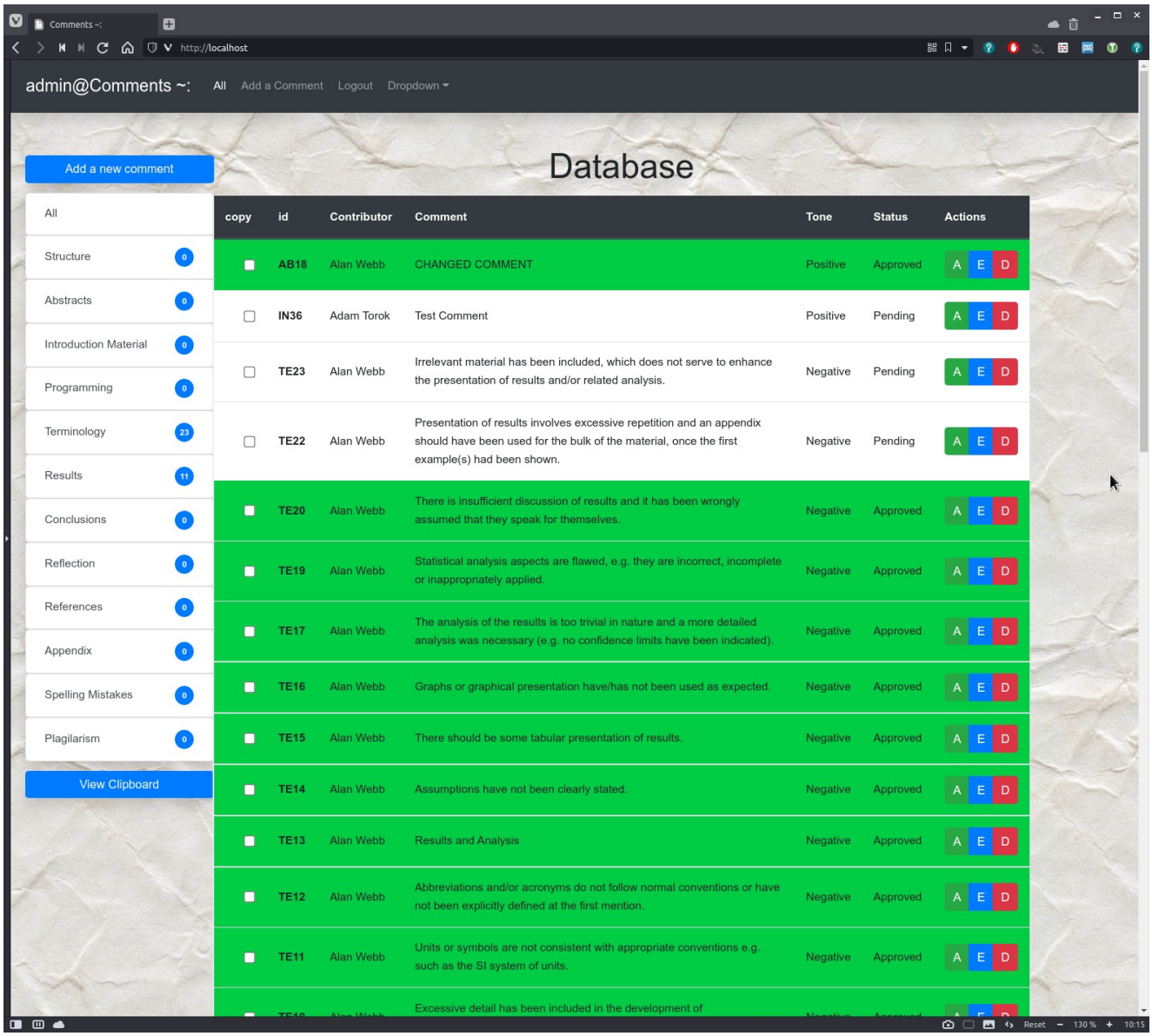


Figure 10: Comment has been deleted

USER

As expected, Comment **AB18** has changed, comment **TE21** has been deleted on the user side.

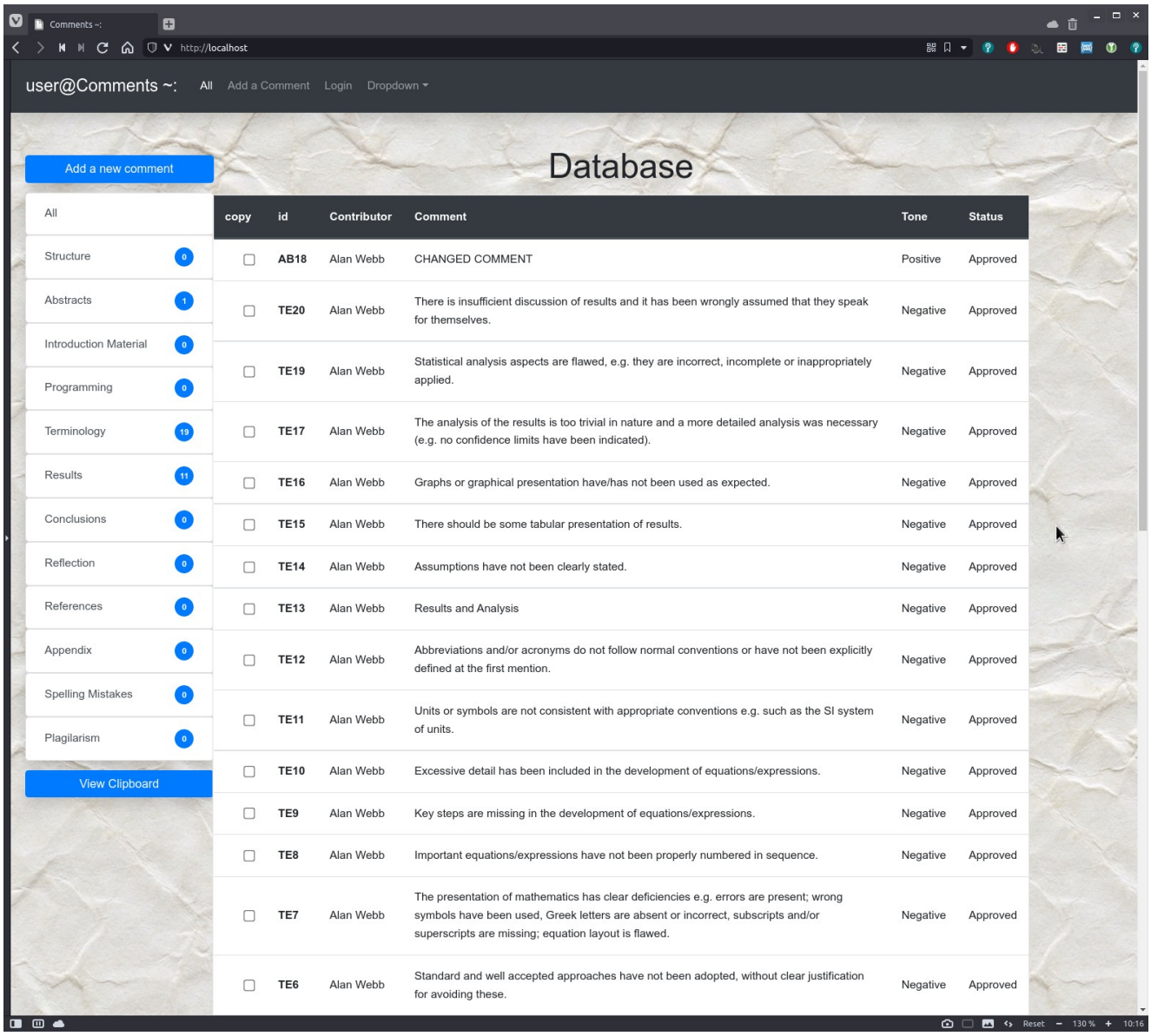


Figure 11: Result

CLIPBOARD

Both usertype can use this functionality. The checked checkboxes text will be gathered and rendered in a modal. With the copy button we can cipy the contents to our clipboard.

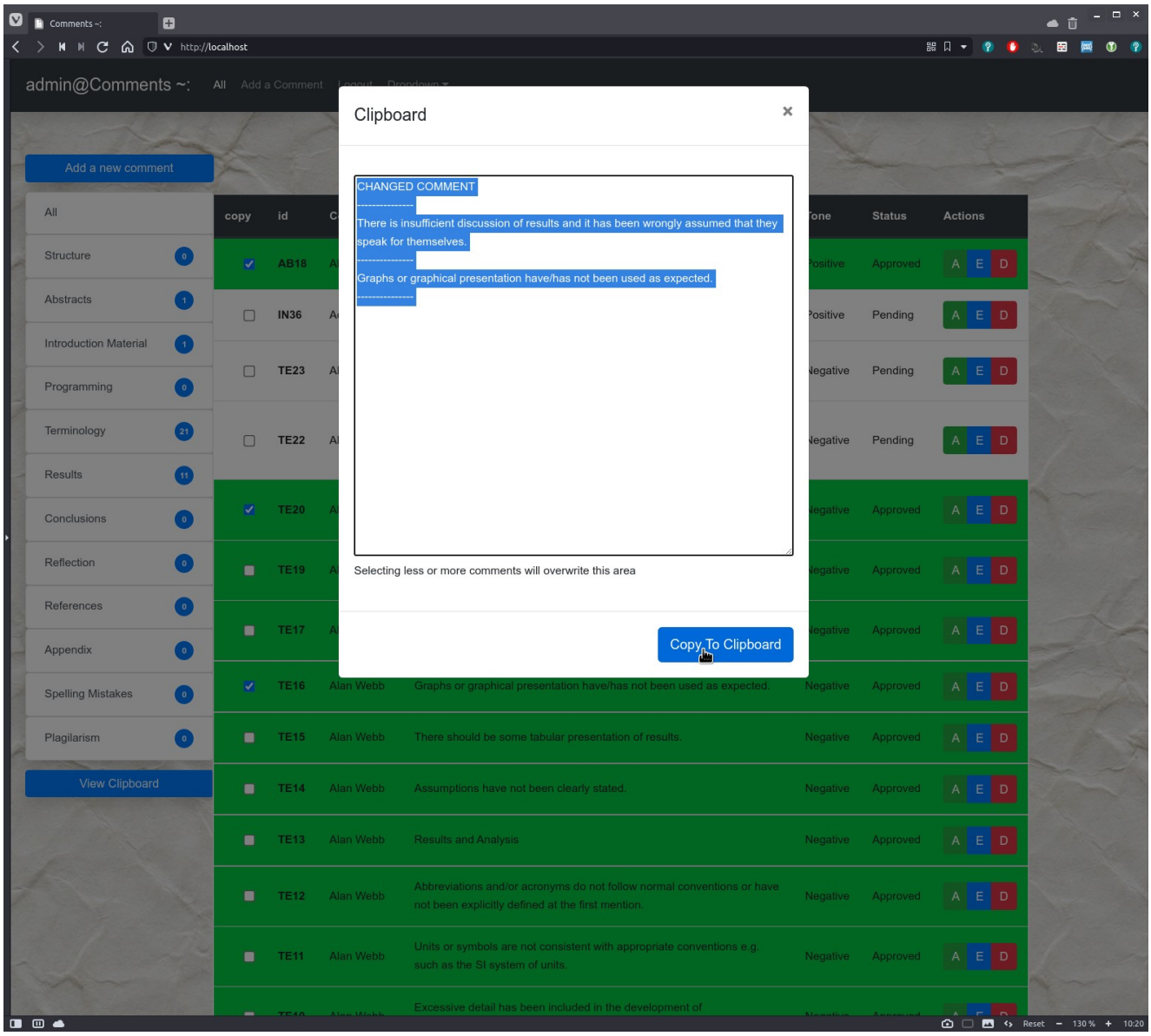


Figure 12: Clipboard