

APPENDIX E

List of Prompts Used

Rubric for List of prompts used

1.

<https://chatgpt.com/share/679773fe-d768-8001-9854-43780840918c>

2.

<https://chatgpt.com/share/679774d2-c25c-8001-a942-15bb13f416c3>

3.

<https://chatgpt.com/share/6797759a-2a40-8001-b110-1b6ba010779c>

4.

<https://chatgpt.com/share/679775b4-afd8-8001-bbf9-29016b1cfdbb>

5.

<https://chatgpt.com/share/679775d0-63d4-8001-89a0-3ac69aad0d99>

6.

<https://chatgpt.com/share/6797767b-b464-8001-90bf-d963f2dd03fb>

APPENDIX F

Screenshots inside files and interfaces

```
JS main.js  X
control > JS main.js > ...
1  const express = require('express');
2  const app = express();
3  const path = require('path');
4  const hbs = require('hbs');
5  const session = require('express-session');
6  const moment = require('moment');
7
8
9  hbs.registerHelper('inc', value => parseInt(value) + 1);
10 hbs.registerHelper('substring', (str, start, end) => typeof str === 'string' ? str.substring(start, end) : '');
11 hbs.registerHelper('eq', (a, b) => a === b);
12 hbs.registerHelper('formatDate', (date, format) => (!date ? '' : moment(date).format(format)));
13
14 app.set('view engine', 'hbs');
15 app.set('views', path.join(__dirname, '../view'));
16 app.use(express.static(path.join(__dirname, '../view/public')));
17 app.use(express.static(path.join(__dirname, '../model/Files')));
18 app.use('/uploads', express.static(path.join(__dirname, '../uploads')));
19
20 app.use(express.urlencoded({ extended: true }));
21 app.use(express.json());
22 app.use(session({
23   secret: '1234',
24   resave: false,
25   saveUninitialized: true
26 }));
27
28
```

main.js (1/4)

```
28 const collection = require('../model/Users');
29
30
31
32 app.get('/Login', (req, res) => {
33   res.render('LoginView');
34 });
35 app.post('/login', async (req, res) => {
36   try {
37     const check = await collection.findOne({ username: req.body.username, password: req.body.password });
38     if (check.username === 'admin' && check.password === '123') {
39       return res.redirect('/Admin');
40     } else if (check.password === req.body.password) {
41       req.session.username = check.username;
42       res.redirect('/Home');
43     } else {
44       res.render('LoginView', { error: 'Invalid username or password' });
45     }
46   } catch {
47     res.render('LoginView', { error: 'Invalid username or password' });
48   }
49 });
50
```

main.js (2/4)

```

app.get('/register', (req, res) => {
  res.render('RegisterView');
});
app.post('/register', async (req, res) => {
  try {
    const existingUser = await collection.findOne({ username: req.body.username });
    if (existingUser) {
      return res.send(`

        <h1>Registration Failed</h1>
        <p>The username <strong>${req.body.username}</strong> is already taken. Please choose a different username</p>
        <a href="/register">Go back to Register</a>
      `);
    }
    await collection.create({
      username: req.body.username,
      password: req.body.password,
      email: req.body.email,
      phone: req.body.phone,
    });
    res.redirect('/Login');
  } catch (error) {
    console.error('Error during registration:', error);
    res.status(500).send('An error occurred during registration.');
  }
});

```

main.js (3/4)

```

78  const CategoryController = require('./CategoryController');
79  const FileController = require('./FileController');
80  const PostController = require('./PostController');
81  const RepliesController = require('./RepliesController');
82  const UserController = require('./UserController');
83  const SearchController = require('./SearchController');
84  const EditUploadsController = require('./EditUploadsController');

85
86
87  app.use(CategoryController);
88  app.use(FileController);
89  app.use(PostController);
90  app.use(RepliesController);
91  app.use(UserController);
92  app.use(SearchController);
93  app.use(EditUploadsController);

94
95
96
97  app.get('/Home', (req, res) => {
98    res.render('HomeView');
99  });
100 app.get('/Admin', (req, res) => {
101   res.render('AdminView');
102 });
103
104 app.listen(3000, function () {
105   console.log('Server is running');
106 });

```

main.js (4/4)

```
JS CategoryController.js ×
control > JS CategoryController.js > ...
1 const express = require('express');
2 const router = express.Router();
3 const collectionC = require('../model/categories');
4
5 router.get('/Categories', async (req, res) => {
6   const categories = await collectionC.find().collation({ locale: 'en', strength: 2 }).sort({ Categoryname: 1 });
7   res.render('CategoryView', { categories });
8 });
9 router.post('/Categories', async (req, res) => {
10   try {
11     const existingCategory = await collectionC.findOne({ Categoryname: req.body.Categoryname });
12     if (existingCategory) {
13       return res.status(400).json({ error: 'Category already exists' });
14     }
15     await collectionC.create({ Categoryname: req.body.Categoryname });
16     res.redirect('/Categories');
17   } catch (error) {
18     res.status(500).send('An error occurred while adding the category.');
19   }
20 });
21 router.post('/Categories/delete/:id', async (req, res) => {
22   try {
23     await collectionC.findByIdAndDelete(req.params.id);
24     res.redirect('/Categories');
25   } catch (error) {
26     res.status(500).send('An error occurred while deleting the category.');
27   }
28 });
29
30 module.exports = router;
```

CategoryController.js

```
JS EditUploadsController.js ×
control > JS EditUploadsController.js > ...
1 const express = require('express');
2 const router = express.Router();
3 const collectionF = require('../model/files');
4 const collectionP = require('../model/Posts');
5 const collectionC = require('../model/categories');
6
7
8
9
10 router.get('/EditUploads', async (req, res) => {
11   const files = await collectionF.find({ uploadedBy: req.session.username });
12   const posts = await collectionP.find({ UploadedBy: req.session.username });
13   res.render('EditUploadsView', { files, posts });
14 });
15
16 router.post('/delete/file/:id', async (req, res) => {
17   try {
18     await collectionF.findByIdAndDelete(req.params.id);
19     res.redirect('/EditUploads');
20   } catch (error) {
21     res.status(500).send('An error occurred while deleting the file.');
22   }
23 });
24
25
26 router.post('/delete/post/:id', async (req, res) => {
27   try {
28     await collectionP.findByIdAndDelete(req.params.id);
29     res.redirect('/EditUploads');
30   } catch (error) {
31     res.status(500).send('An error occurred while deleting the post.');
32   }
33 });
34
```

EditUploadsController.js (1/3)

```

js EditUploadsController.js ×
control > js EditUploadsController.js > ...
35   router.get('/edit/file/:id', async (req, res) => {
36     try {
37       const file = await collectionF.findById(req.params.id);
38       if (!file) return res.status(404).send('File not found');
39       const categories = await collectionC.find().collation({ locale: 'en', strength: 2 }).sort({ Categoryname: 1 });
40       res.render('EditView', { file, categories, username: req.session.username, type: 'file' });
41     } catch (error) {
42       res.status(500).send('An error occurred while fetching the file.');
43     }
44   });
45
46 router.post('/edit/file/:id', async (req, res) => {
47   try {
48     const updateData = {
49       title: req.body.title,
50       description: req.body.description,
51       Category: req.body.Category
52     };
53     await collectionF.findByIdAndUpdate(req.params.id, updateData);
54     res.redirect('/EditUploads');
55   } catch (error) {
56     res.status(500).send('An error occurred while updating the file.');
57   }
58 });
59
60 router.get('/edit/post/:id', async (req, res) => {
61   try {
62     const post = await collectionP.findById(req.params.id);
63     if (!post) return res.status(404).send('Post not found');
64     const categories = await collectionC.find().collation({ locale: 'en', strength: 2 }).sort({ Categoryname: 1 });
65     res.render('EditView', { post, categories, username: req.session.username, type: 'post' });
66   } catch (error) {
67     res.status(500).send('An error occurred while fetching the post.');
68   }
69 });

```

EditUploadsController.js (2/3)

```

70
71 router.post('/edit/post/:id', async (req, res) => {
72   try {
73     const updateData = {
74       Title: req.body.Title,
75       Description: req.body.Description,
76       Category: req.body.Category
77     };
78     await collectionP.findByIdAndUpdate(req.params.id, updateData);
79     res.redirect('/EditUploads');
80   } catch (error) {
81     res.status(500).send('An error occurred while updating the post.');
82   }
83 });
84
85 module.exports = router;

```

EditUploadsController.js (3/3)

```

control > js FileController.js > ...
1  const express = require('express');
2  const router = express.Router();
3  const collectionF = require('../model/Files');
4  const collectionC = require('../model/categories');
5  const multer = require('multer');
6  const path = require('path');
7
8  const storage = multer.diskStorage({
9    destination: function (req, file, cb) {
10      cb(null, path.resolve(__dirname, '../uploads'));
11    },
12    filename: function (req, file, cb) {
13      cb(null, Date.now() + '-' + file.originalname);
14    }
15  });
16  function fileFilter(req, file, cb) {
17    const allowedTypes = ['.pdf', '.zip'];
18    const ext = path.extname(file.originalname).toLowerCase();
19    if (allowedTypes.includes(ext)) {
20      cb(null, true);
21    } else {
22      cb(new Error('Only PDF and ZIP files are allowed!'), false);
23    }
24  }
25  const upload = multer({ storage: storage, fileFilter: fileFilter });
26
27
28  router.get('/Upload', async (req, res) => {
29    const categories = await collectionC.find().collation({ locale: 'en', strength: 2 }).sort({ Categoryname: 1 });
30    res.render('UploadView', { categories });
31  });

```

FileController.js (1/4)

```

js FileController.js X
control > js FileController.js > ...
32  router.post('/Upload', upload.single('file'), async (req, res) => {
33    try {
34      if (!req.file) return res.status(400).send('No file uploaded.');
35      await collectionF.create({
36        title: req.body.title,
37        description: req.body.description,
38        file: req.file.filename,
39        Category: req.body.Category,
40        uploadedBy: req.session.username,
41        status: 'verifying'
42      });
43      res.redirect('/Materials');
44    } catch (error) {
45      res.status(500).send('An error occurred during file upload.');
46    }
47  });
48
49  router.get('/Materials', async (req, res) => {
50    const sort = req.query.sort || 'category';
51    let Files = await collectionF.find();
52    let grouped = {};
53    if (sort === 'alphabetic') {
54      Files.forEach(file => {
55        const letter = file.Title ? file.Title[0].toUpperCase() : '#';
56        if (!grouped[letter]) grouped[letter] = [];
57        grouped[letter].push(file);
58      });
59      grouped = Object.fromEntries(Object.entries(grouped).sort());
60    } else if (sort === 'category') {
61      Files.forEach(Files => {
62        const cat = Files.Category || 'Uncategorized';
63        if (!grouped[cat]) grouped[cat] = [];
64        grouped[cat].push(Files);
65      });
66  });

```

FileController.js (2/4)

```

66   | grouped = Object.fromEntries(Object.entries(grouped).sort());
67   |
68   | res.render('AllMaterialsView', { grouped, username: req.session.username, sort });
69   });
70
71 router.get('/MyMaterials', async (req, res) => {
72   const files = await collectionF.find({ uploadedBy: req.session.username });
73   res.render('MyMaterialsView', { files });
74 });
75
76 router.get('/FilesLists', async (req, res) => {
77   const sort = req.query.sort || 'date';
78   let files = await collectionF.find();
79   if (sort === 'alphabetic') {
80     files = files.sort((a, b) => (a.title || '').localeCompare(b.title || ''));
81   } else if (sort === 'category') {
82     files = files.sort((a, b) => (a.Category || '').localeCompare(b.Category || ''));
83   } else if (sort === 'date') {
84     files = files.sort((a, b) => {
85       const dateA = a.createdAt || a._id.getTimestamp();
86       const dateB = b.createdAt || b._id.getTimestamp();
87       return dateB - dateA;
88     });
89   }
90   res.render('manageFileView', { files, sort });
91 });
92 router.post('/FilesLists/delete/:id', async (req, res) => {
93   try {
94     await collectionF.findByIdAndUpdateAndDelete(req.params.id);
95     res.redirect('/FilesLists');
96   } catch (error) {
97     res.status(500).send('An error occurred while deleting the file.');
98   }
99 });

```

FileController.js (3/4)

```

control > JS FileController.js ...
101  router.get('/viewFile/:id', async (req, res) => {
102    try {
103      const file = await collectionF.findById(req.params.id);
104      if (!file) return res.status(404).send('File not found');
105      res.render('FileDetailsView', { file, username: req.session.username });
106    } catch (error) {
107      res.status(500).send('An error occurred while fetching the file.');
108    }
109  });
110
111 router.get('/verifications', async (req, res) => {
112   const sort = req.query.sort || 'date';
113   let files = await collectionF.find({ status: 'verifying' });
114   if (sort === 'alphabetic') {
115     files = files.sort((a, b) => (a.title || '').localeCompare(b.title || ''));
116   } else if (sort === 'category') {
117     files = files.sort((a, b) => (a.Category || '').localeCompare(b.Category || ''));
118   } else if (sort === 'date') {
119     files = files.sort((a, b) => {
120       const dateA = a.createdAt || a._id.getTimestamp();
121       const dateB = b.createdAt || b._id.getTimestamp();
122       return dateB - dateA;
123     });
124   }
125   res.render('Verification', { files, sort });
126 });
127 router.post('/verifications/accept/:id', async (req, res) => {
128   await collectionF.findByIdAndUpdateAndUpdate(req.params.id, { status: 'accepted' });
129   res.redirect('/verifications');
130 });
131 router.post('/verifications/decline/:id', async (req, res) => {
132   await collectionF.findByIdAndUpdateAndUpdate(req.params.id, { status: 'declined' });
133   res.redirect('/verifications');
134 });
135

```

FileController.js (4/4)

```
JS PostController.js ×
control > JS PostController.js > ...
1  |const express = require('express');
2  |const router = express.Router();
3  |const collectionP = require('../model/Posts');
4  |const collectionC = require('../model/categories');
5  |const collectionR = require('../model/Replies');
6
7  |router.get('/PostsLists', async (req, res) => {
8  |  const sort = req.query.sort || 'date';
9  |  let posts = await collectionP.find();
10 |  if (sort === 'alphabetic') {
11 |    posts = posts.sort((a, b) => (a.Title || '').localeCompare(b.Title || ''));
12 |  } else if (sort === 'category') {
13 |    posts = posts.sort((a, b) => (a.Category || '').localeCompare(b.Category || ''));
14 |  } else if (sort === 'date') {
15 |    posts = posts.sort((a, b) => {
16 |      const dateA = a.createdAt || a._id.getTimestamp();
17 |      const dateB = b.createdAt || b._id.getTimestamp();
18 |      return dateB - dateA;
19 |    });
20 |  }
21 |  res.render('managePostView', { posts, sort });
22 |});
23 |router.post('/Posts/delete/:id', async (req, res) => {
24 |  try {
25 |    await collectionP.findByIdAndDelete(req.params.id);
26 |    res.redirect('/PostsLists');
27 |  } catch (error) {
28 |    res.status(500).send('An error occurred while deleting the post.');
29 |  }
30 |});
31 |;
```

PostController.js (1/3)

```
32 |router.get('/Discussions', async (req, res) => {
33 |  const sort = req.query.sort || 'category';
34 |  let posts = await collectionP.find();
35 |  let grouped = {};
36 |  if (sort === 'alphabetic') {
37 |    posts.forEach(post => {
38 |      const letter = post.Title ? post.Title[0].toUpperCase() : '#';
39 |      if (!grouped[letter]) grouped[letter] = [];
40 |      grouped[letter].push(post);
41 |    });
42 |    grouped = Object.fromEntries(Object.entries(grouped).sort());
43 |  } else if (sort === 'category') {
44 |    posts.forEach(post => {
45 |      const cat = post.Category || 'Uncategorized';
46 |      if (!grouped[cat]) grouped[cat] = [];
47 |      grouped[cat].push(post);
48 |    });
49 |    grouped = Object.fromEntries(Object.entries(grouped).sort());
50 |  }
51 |  res.render('DiscussionView', { grouped, username: req.session.username, sort });
52 |});
```

PostController.js (2/3)

```

53
54   router.get('/Create', async (req, res) => {
55     const categories = await collectionC.find().collation({ locale: 'en', strength: 2 }).sort({ Categoryname: 1 });
56     res.render('CreatePostView', { categories });
57   });
58   router.post('/Create', async (req, res) => {
59     try {
60       const username = req.session.username;
61       if (!username) return res.status(401).send('You must be logged in to create a post.');
62       await collectionP.create({
63         Title: req.body.Title,
64         Description: req.body.Description,
65         Category: req.body.Category,
66         UploadedBy: username,
67       });
68       res.redirect('/Discussions');
69     } catch (error) {
70       res.status(500).send('An error occurred during file upload.');
71     }
72   });
73
74   router.get('/view/:id', async (req, res) => {
75     try {
76       const post = await collectionP.findById(req.params.id);
77       const replies = await collectionR.find({ postId: post._id });
78       if (!post) return res.status(404).send('Post not found');
79       res.render('PostDetailsView', { post, username: req.session.username, replies });
80     } catch (error) {
81       res.status(500).send('An error occurred while fetching the post.');
82     }
83   });
84
85   module.exports = router;

```

PostController.js (3/3)

```

JS RepliesController.js X
control > JS RepliesController.js > ...
1  const express = require('express');
2  const router = express.Router();
3  const collectionR = require('../model/Replies');
4
5  router.post('/reply/:postId', async (req, res) => {
6    try {
7      const username = req.session.username;
8      await collectionR.create({
9        Reply: req.body.Reply,
10        ReplyBy: username,
11        postId: req.params.postId
12      });
13      res.redirect('/view/' + req.params.postId);
14    } catch (error) {
15      res.status(500).send('An error occurred while saving the reply.');
16    }
17  });
18
19  module.exports = router;

```

RepliesController.js

```
JS SearchController.js ×
control > JS SearchController.js > ...
1  const express = require('express');
2  const router = express.Router();
3  const Fuse = require('fuse.js');
4  const collectionP = require('../model/Posts');
5  const collectionF = require('../model/Files');
6  const collectionC = require('../model/categories');
7
8  router.get('/search', async (req, res) => {
9    const query = req.query.q || '';
10   const selectedCategory = req.query.category || '';
11   let allPosts = await collectionP.find();
12   let allMaterials = await collectionF.find();
13   if (selectedCategory) {
14     allPosts = allPosts.filter(post => post.Category === selectedCategory);
15   }
16   let posts = allPosts;
17   if (query) {
18     const fusePosts = new Fuse(allPosts, { keys: ['Title', 'Description'], threshold: 0.4 });
19     posts = fusePosts.search(query).map(r => r.item);
20   }
21   let materials = allMaterials;
22   if (query) {
23     const fuseMaterials = new Fuse(allMaterials, { keys: ['title', 'description', 'Category'], threshold: 0.4 });
24     materials = fuseMaterials.search(query).map(r => r.item);
25   }
26   const categories = await collectionC.find().collation({ locale: 'en', strength: 2 }).sort({ Categoryname: 1 });
27   res.render('SearchResultView', { posts, categories, query, selectedCategory, username: req.session.username });
28 });
29
30 module.exports = router;
```

SearchController.js

```
JS UserController.js ×
control > JS UserController.js > ...
1  const express = require('express');
2  const router = express.Router();
3  const collection = require('../model/Users');
4
5  router.get('/Userlists', async (req, res) => {
6    res.render('ManageUserView', { users: await collection.find() });
7  );
8  router.post('/Userlists/delete/:id', async (req, res) => {
9    try {
10      await collection.findByIdAndDelete(req.params.id);
11      res.redirect('/Userlists');
12    } catch (error) {
13      res.status(500).send('An error occurred while deleting the user.');
14    }
15  );
16
17 module.exports = router;
```

UserController.js

```
JS categories.js ×
model > JS categories.js > ...
1  const mongoose = require('mongoose');
2
3  mongoose.connect('mongodb://localhost:27017/ermad')
4
5
6  const CategorySchema = new mongoose.Schema({
7    CategoryID: {
8      type: Number,
9      required: true,
10     unique: true,
11     default: 0
12   },
13   Categoryname: {
14     type: String,
15     required: true
16   }
17 });
18
19 CategorySchema.pre('save', async function (next) {
20   if (this.isNew) {
21     const lastCategory = await mongoose.model("Categories").findOne().sort({ CategoryID: -1 });
22     this.CategoryID = lastCategory ? lastCategory.CategoryID + 1 : 1;
23   }
24   next();
25 });
26
27 const collectionC = new mongoose.model("Categories", CategorySchema);
28
29 module.exports = collectionC;
```

Categories.js

```
JS Files.js ×
model > JS Files.js > ...
1  const mongoose = require('mongoose');
2
3  const fileSchema = new mongoose.Schema({
4    file: {
5      type: String,
6      required: true,
7    },
8    title: {
9      type: String,
10     required: true,
11   },
12   description: {
13     type: String,
14     required: true,
15   },
16   Category: {
17     type: String,
18     required: true
19   },
20   date: {
21     type: Date,
22     default: Date.now,
23   },
24   uploadedBy: {
25     type: String,
26     required: true,
27   },
28   status: {
29     type: String,
30     default: 'verifying'
31   }
32 });
33
34 const collectionF = mongoose.model('Files', fileSchema);
35 module.exports = mongoose.model('Files', fileSchema);
```

Files.js

```
JS Posts.js  X
model > JS Posts.js > ...
1  const mongoose = require('mongoose');
2
3  mongoose.connect('mongodb://localhost:27017/ermad')
4
5  const PosSchema = new mongoose.Schema({
6
7      Title: {
8          type: String,
9          required: true
10     },
11     Description: {
12         type: String,
13         required: true
14     },
15     Category: {
16         type: String,
17         required: true
18     },
19     UploadDate: {
20         type: Date,
21         default: Date.now
22     },
23     UploadedBy: {
24         type: String,
25         required: true
26     },
27 });
28 });
29
30 const collectionP = new mongoose.model("Posts", PosSchema);
31
32 module.exports = collectionP;
```

Posts.js

```
JS Replys.js  X
model > JS Replys.js > ...
1  const mongoose = require('mongoose');
2
3  mongoose.connect('mongodb://localhost:27017/ermad')
4
5
6
\ERMADE\models\Files.js> Schema = new mongoose.Schema({
7      Reply: {
8          type: String,
9          required: true
10     },
11     ReplyDate: {
12         type: Date,
13         default: Date.now
14     },
15     ReplyBy: {
16         type: String,
17         required: true
18     },
19     postId: {
20         type: String,
21         required: true
22     },
23 });
24 });
25
26 const collectionR = new mongoose.model("Replys", Schema);
27
28 module.exports = collectionR;
```

Replys.js

```

JS Users.js  X
model > JS Users.js > [e] RegisterSchemaa
  6  const RegisterSchemaa = new mongoose.Schema({
  7    UserId: {
  8      type: Number,
  9      required: true,
 10      unique: true,
 11      default: 0
 12    },
 13    username: {
 14      type: String,
 15      required: true
 16    },
 17    password: {
 18      type: String,
 19      required: true
 20    },
 21    email: {
 22      type: String,
 23      required: true
 24    },
 25    phone: {
 26      type: Number,
 27      required: true
 28    }
 29  });
 30
 31 RegisterSchemaa.pre('save', async function (next) {
 32   if (this.isNew) {
 33     const lastUser = await mongoose.model("Users").findOne().sort({ UserId: -1 });
 34     this.UserId = lastUser ? lastUser.UserId + 1 : 1;
 35   }
 36   next();
 37 });
 38
 39 const collection = new mongoose.model("Users", RegisterSchemaa);
 40 module.exports = collection;

```

Users.js

```

view > ~ AdminView.hbs > ...
  1  <!DOCTYPE html>
  2  <html lang="en">
  3  <head>
  4    <meta charset="UTF-8">
  5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  6    <title>ERMAD - Home</title>
  7    <link rel="stylesheet" href="style.css">
  8  </head>
  9  <body>
10    <header class="header">
11      <h1 class="logo"><img src ="ERMADheader.png"></h1>
12      <nav class="user-nav">
13        <span style="margin-right: 10px;">
14          {{#if username}}Hello, {{username}}{{/if}}
15        </span>
16        <a href="/login" class="logout-btn">LOGOUT</a>
17      </nav>
18    </header>
19
20    <main class="main-home">
21      <div class="Main-img">
22        
23      </div>
24      <nav class="quick-links">
25        <a href="/Userlists" class="btn">Manage Users</a>
26        <a href="/PostsLists" class="btn">Manage Posts</a>
27        <a href="/filesLists" class="btn">Manage Files</a>
28        <a href="/Verifications" class="btn">Verify files</a>
29        <a href="/Categories" class="btn">Manage Categories</a>
30      </nav>
31    </main>
32
33    <script src="..../control/main.js"></script>
34  </body>
35 </html>

```

AdminView.hbs

```

view > AllMaterialsView.hbs > html > head > title
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>ERMAD - All Materials</title>
7      <link rel="stylesheet" href="/style.css">
8  </head>
9  <body>
10     <header class="header">
11         <h1 class="logo"><img src = "ERMADheader.png"></h1>
12         <div class="search-container">
13             <form action="/search" method="GET">
14                 <input type="text" name="q" placeholder="Search ..." >
15                 <select name="category">
16                     <option value="">All Categories</option>
17                     {{#each categories}}
18                         <option value="{{this.Categoryname}}">{{this.Categoryname}}</option>
19                     {{/each}}
20                 </select>
21                 <button type="submit" >Search</button>
22             </form>
23         </div>
24         <nav class="user-nav">
25             <span class="displayUser">
26                 {{#if username}}Hello, {{username}}{{/if}}
27             </span>
28             <a href="/login" class="logout-btn">LOGOUT</a>
29         </nav>
30     </header>
31     <div class="back-button-container">
32         <button onclick="window.location.href='/home'" class="back-button">
33             <svg class="arrow-icon" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2">
34                 <path d="M5 12h14M12 5l7 7 7 7"/>
35             </svg>

```

AllMaterialsView.hbs (1/3)

```

36     <div class="back-button-container">
37         <button>
38             <form id="sort-form" method="GET" action="/Materials" class="sort-form">
39                 <label for="sort-select">Sort by:</label>
40                 <select id="sort-select" name="sort" onchange="this.form.submit()">
41                     <option value="category" {{#if (eq sort 'category')}}selected{{/if}}>Category</option>
42                     <option value="alphabetic" {{#if (eq sort 'alphabetic')}}selected{{/if}}>Alphabetic</option>
43                 </select>
44             </form>
45         </div>
46         <div class="front-button">
47             <nav class="quick-links">
48                 <div class="upload-dropdown">
49                     <button onclick="toggleUploadMenu()" class="btn">Upload</button>
50                     <div id="uploadMenu">
51                         <a href="/Upload">Upload Material</a>
52                         <a href="/Create">Create Post</a>
53                     </div>
54                 </div>
55             </nav>
56             <script>
57                 function toggleUploadMenu() {
58                     var menu = document.getElementById('uploadMenu');
59                     menu.style.display = (menu.style.display === 'block') ? 'none' : 'block';
60                     // Hide menu when clicking outside
61                     document.addEventListener('click', function handler(e) {
62                         if (!menu.contains(e.target) && e.target.className !== 'btn') {
63                             menu.style.display = 'none';
64                             document.removeEventListener('click', handler);
65                         }
66                     });
67                 }

```

AllMaterialsView.hbs (2/3)

```

71   <section class="discussion-list">
72     {{#each grouped}}
73       <h2 class="group-title">{{@key}}</h2>
74       <div class="grid-container">
75         {{#each this}}
76           {{#if (eq this.status "accepted")}}
77             <div class="discussion-item">
78               <a href="#">{{this._id}}</a> class="discussion-link">
79                 <h3 class="discussion-title">{{this.title}}</h3>
80                 <div class="discussion-preview">
81                   {{#if this.Preview}}
82                     
83                   {{else}}
84                     <span class="no-preview">No preview available</span>
85                   {{/if}}
86                 </div>
87                 <p class="discussion-description">
88                   {{#if this.description}}
89                     {{substring this.description 0 50}}...
89                   {{/if}}
90                 </p>
91               </a>
92             </div>
93           {{/if}}
94         {{/each}}
95       </div>
96       <div class="line"></div>
97     {{/each}}
98   </section>
99 </main>

```

AllMaterialsView.hbs (3/3)

```

1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>ERMAD - Manage Categories</title>
7      <link rel="stylesheet" href="style.css">
8    </head>
9    <body>
10      <header class="header">
11        <h1 class="logo"><img src = "ERMADheader.png"></h1>
12        <nav class="user-nav">
13          <span style="margin-right: 10px;">
14            {{#if username}}Hello, {{username}}{{/if}}
15          </span>
16          <a href="/login" class="logout-btn">LOGOUT</a>
17        </nav>
18      </header>
19      <h1>Manage Categories </h1>
20      <div class="back-button-container">
21        <button onclick="window.location.href='/Admin'" class="back-button">
22          <svg class="arrow-icon" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2">
23            <path d="M5 12h14M12 5l7 7-7 7"/>
24          </svg>
25        </button>
26        <form id="upload-form" action="/Categories" method="POST">
27          <label for="Categoryname">Category Name:</label>
28          <input type="text" id="Categoryname" name="Categoryname" required>
29          <button type="submit" class="btn" >Add Category</button>
30        </form>
31      </div>
32      <main class="main-content">
33

```

CategoryView.hbs (1/2)

```

34      <table>
35        <thead>
36          <tr>
37            <th>ID</th>
38            <th>Category Name</th>
39            <th>Number of Materials</th>
40            <th>Actions</th>
41          </tr>
42        </thead>
43        <tbody>
44          {{#each categories}}
45            <tr>
46              <td>{{inc @index}}</td>
47              <td>{{this.Categoryname}}</td>
48              <td>{{this.materialCount}}</td>
49              <td>
50                <form action="/Categories/delete/{{this._id}}" method="POST" onsubmit="return confirmDelete();">
51                  <button type="submit" class="delete-btn">Delete</button>
52                </form>
53              </td>
54            </tr>
55          {{/each}}
56        </tbody>
57      </table>
58    </main>
59    <script>
60      function confirmDelete() {
61        return confirm('Are you sure you want to delete this Category?');
62        if (confirmDelete()) {
63          return confirm('Delete the category will cause the deletion of all materials inside, do you want to');
64        } else {
65          return false;
66        }
67    </script>

```

CategoryView.hbs (2/2)

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>ERMAD - Create Post</title>
7      <link rel="stylesheet" href="style.css">
8  </head>
9  <body>
10     <header class="header">
11         <h1 class="logo"><img src = "ERMADheader.png"></h1>
12         <div class="search-container">
13             <form action="/search" method="GET">
14                 <input type="text" name="q" placeholder="Search ..." >
15                 <select name="category">
16                     <option value="">All Categories</option>
17                     {{#each categories}}
18                         <option value="{{this.Categoryname}}>{{this.Categoryname}}</option>
19                     {{/each}}
20                 </select>
21                 <button type="submit" >Search</button>
22             </form>
23         </div>
24         <nav class="user-nav">
25             <span style="margin-right: 10px;">
26                 {{#if username}}Hello, {{username}}{{/if}}
27             </span>
28             <a href="/login" class="logout-btn">LOGOUT</a>
29         </nav>
30     </header>
31     <div class="back-button-container">
32         <button onclick="window.location.href='/Discussions'" class="back-button">
33             <svg class="arrow-icon" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="none" stroke="cun
34             <path d="M5 12h14M12 5l7 7 7 7"/>
35         </svg>

```

CreatePostView.hbs (1/2)

```

39     <main class="main-content">
40         <section class="upload-section">
41             <h2>Create Post</h2>
42             <form id="upload-form" action="/Create" method="POST">
43                 <div class="form-group">
44                     <label for="Title">Title</label>
45                     <input type="text" id="Title" name="Title" required>
46                 </div>
47                 <div class="form-group">
48                     <label for="Description">Description</label>
49                     <textarea id="Description" name="Description" required></textarea>
50                 </div>
51                 <div class="form-group">
52                     <label for="Category">Category</label>
53                     <select id="Category" name="Category" required>
54                         {{#if categories}}
55                         {{#each categories}}
56                             <option value="{{this.Categoryname}}>{{this.Categoryname}}</option>
57                         {{/each}}
58                         {{else}}
59                             <option disabled>No categories available</option>
60                         {{/if}}
61                     </select>
62                 </div>
63                 <button type="submit" class="btn">Upload</button>
64             </form>
65         </section>
66     </main>
67

```

CreatePostView.hbs (2/2)

```

view > DiscussionView.hbs > html > head > title
2   <html lang="en">
9   <body>
31     <div class="back-button-container">
32       <button onclick="window.location.href='/home'" class="back-button">
33         <svg class="arrow-icon" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="none" stroke="curr
34           | path d="M5 12h14M12 5l7 -7 7 />
35         </svg>
36       </button>
37       <form id="sort-form" method="GET" action="/Discussions" class="sort-form">
38         <label for="sort-select">Sort by:</label>
39         <select id="sort-select" name="sort" onchange="this.form.submit()">
40           <option value="category" {{#if (eq sort 'category')}}selected{{/if}}>Category</option>
41           <option value="Alphabetic" {{#if (eq sort 'Alphabetic')}}selected{{/if}}>Alphabetic</option>
42         </select>
43       </form>
44     </div>
45
46     <div class="front-button">
47       <nav class="quick-links">
48         <div class="upload-dropdown">
49           <button onclick="toggleUploadMenu()" class="btn">Upload</button>
50           <div id="uploadMenu">
51             <a href="/Upload">Upload Material</a>
52             <a href="/Create">Create Post</a>
53           </div>
54         </div>
55       </nav>

```

DiscussionView.hbs (1/2)

```

57   function toggleUploadMenu() {
58     var menu = document.getElementById('uploadMenu');
59     menu.style.display = (menu.style.display === 'block') ? 'none' : 'block';
60     // Hide menu when clicking outside
61     document.addEventListener('click', function handler(e) {
62       if (!menu.contains(e.target) && e.target.className !== 'btn') {
63         menu.style.display = 'none';
64         document.removeEventListener('click', handler);
65       }
66     });
67   }
68   </script>
69 </div>
70 <main>
71   <section class="discussion-list">
72     {{#each grouped}}
73       <h2 class="group-title">{{@key}}</h2>
74       <div class="grid-container">
75         {{#each this}}
76           <div class="discussion-item">
77             <a href="/view/{{this._id}}" class="discussion-link">
78               <h3 class="discussion-title">{{this.Title}}</h3>
79               <p class="discussion-preview">
80                 {{#if this.Description}}
81                   {{substring this.Description 0 50}}...
82                 {{/if}}
83               </p>
84             </a>
85           </div>
86         {{/each}}
87       </div>
88     {{/each}}
89   </section>
90 </main>
91 </body>

```

DiscussionView.hbs (2/2)

```

view > EditUploadsView.hbs > html > head > title
2   <html lang="en">
9   <body>
31     <div class="back-button-container">
32       <button onclick="window.location.href='/home'" class="back-button">
33         <svg class="arrow-icon" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="none" stroke="curr
34           | path d="M5 12h14M12 5l7 -7 7 />
35         </svg>
36       </button>
37     </div>
38
39     <div class="front-button">
40       <nav class="quick-links">
41         <div class="upload-dropdown">
42           <button onclick="toggleUploadMenu()" class="btn">Upload</button>
43           <div id="uploadMenu">
44             <a href="/Upload">Upload Material</a>
45             <a href="/Create">Create Post</a>
46           </div>
47         </div>
48       </nav>
49       <script>
50         function toggleUploadMenu() {
51           var menu = document.getElementById('uploadMenu');
52           menu.style.display = (menu.style.display === 'block') ? 'none' : 'block';
53           // Hide menu when clicking outside
54           document.addEventListener('click', function handler(e) {
55             if (!menu.contains(e.target) && e.target.className !== 'btn') {
56               menu.style.display = 'none';
57               document.removeEventListener('click', handler);
58             }
59           });
60         }
61       </script>

```

EditUploadsView.hbs (1/2)

```

63   <main>
64     <h2>Files</h2>
65     <table>
66       <thead>
67         <tr>
68           <th>Title</th>
69           <th>Description</th>
70           <th>Category</th>
71           <th>Actions</th>
72         </tr>
73       </thead>
74       <tbody>
75         {{#each files}}
76           <tr>
77             <td>{{this.title}}</td>
78             <td>
79               {{this.description}}
80             </td>
81             <td>{{this.Category}}</td>
82             <td class="actions">
83               <a href="/edit/file/{{this.id}}" class="btn">Edit</a>
84               <form action="/delete/file/{{this.id}}" method="POST" style="display:inline;" onsubmit="
85                 <button type="submit" class="delete-btn">Delete</button>
86               </form>
87             </td>
88           </tr>
89         {{/each}}
90       </tbody>
91     </table>

```

EditUploadsView.hbs (2/2)

```

93   <h2>Posts</h2>
94   <table>
95     <thead>
96       <tr>
97         <th>Title</th>
98         <th>Description</th>
99         <th>Category</th>
100        <th>Actions</th>
101      </tr>
102    </thead>
103    <tbody>
104      {{#each posts}}
105        <tr>
106          <td>{{this.Title}}</td>
107          <td>
108            {{this.Description}}
109          </td>
110          <td>{{this.Category}}</td>
111          <td>
112            <a href="/edit/post/{{this.id}}" class="btn">Edit</a>
113            <form action="/delete/post/{{this.id}}" method="POST" style="display:inline;" onsubmit="
114              <button type="submit" class="delete-btn">Delete</button>
115            </form>
116          </td>
117        </tr>
118      {{/each}}
119    </tbody>
120  </table>

```

EditUploadsView.hbs

```

view > EditView.hbs > HTML > head > title
2   <html lang="en">
9     <body>
10
11       <main class="main-content">
12         <section class="upload-section">
13           {{#if (eq type "file")}}
14             <h2>Edit File</h2>
15             <form id="edit-file-form" action="/edit/file/{{file._id}}" method="POST" enctype="multipart/form-data">
16               <div class="form-group">
17                 <label for="title">Name:</label>
18                 <input type="text" id="title" name="title" value="{{file.title}}" required>
19               </div>
20               <div class="form-group">
21                 <label for="description">Description:</label>
22                 <textarea id="description" name="description" required>{{file.description}}</textarea>
23               </div>
24               <div class="form-group">
25                 <label for="Category">Category:</label>
26                 <select id="Category" name="Category" required>
27                   {{#if categories}}
28                     {{#each categories}}
29                       <option value="{{this.Categoryname}}> {{#if (eq this.Categoryname ../file.Category)}}selected{{/if}}</option>
30                     {{/each}}
31                     {{#else}}
32                       <option disabled>No categories available</option>
33                     {{/if}}
34                   </select>
35                 </div>
36                 <button type="submit" class="btn">Update</button>
37               </form>
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62

```

EditView.hbs (1/2)

```

62     </form>
63     {{else if (eq type "post")}}
64     <h2>Edit Post:</h2>
65     <form id="edit-post-form" action="/edit/post/{{post._id}}" method="POST">
66         <div class="form-group">
67             <label for="Title">Name</label>
68             <input type="text" id="Title" name="Title" value="{{post.Title}}" required>
69         </div>
70         <div class="form-group">
71             <label for="Description">Description</label>
72             <textarea id="Description" name="Description" required>{{post.Description}}</textarea>
73         </div>
74         <div class="form-group">
75             <label for="Category">Category</label>
76             <select id="Category" name="Category" required>
77                 {{#if categories}}
78                 {{#each categories}}
79                     <option value="{{this.Categoryname}}> {{#if (eq this.Categoryname ../post.Category)}>selected</option>
80                     {{/each}}
81                 {{else}}
82                     <option disabled>No categories available</option>
83                 {{/if}}
84             </select>
85         </div>
86         <button type="submit" class="btn">Update</button>
87     </form>
88     {{/if}}
89   </section>
90 </main>
91 
```

EditView.hbs (2/2)

```

38   <main>
39     {{#if (eq file.extension "pdf")}}
40     <iframe src="..uploads/1750753966900-WameedMULecture_2021_92553827.pdf" class="pdf-preview-half"></iframe>
41   {{else}}
42     <div class="pdf-preview-half" style="display:flex;align-items:center;justify-content:center;">
43       <span> preview available</span>
44     </div>
45   {{/if}}
46   <div class="file-details-content">
47     <h1 class="file-title">{{file.title}}</h1>
48     <div class="file-description">{{file.description}}</div>
49     <div class="file-meta">
50       Uploaded by: {{file.uploadedBy}}<br>
51       Date: {{formatDate file.uploadDate "DD/MM/YYYY"}}
52     </div>
53     <div class="download-btn-bottom">
54       <a href="..uploads/{{file.file}}> download class="btn">Download</a>
55     </div>
56   </div>
57 </main>

```

FiledetailsView.hbs

```

view > HomeView.hbs > html > head > title
2   <html lang="en">
9     <body>
10       <div>
11         <nav class="quick-links">
12           <a href="/Materials" class="btn">View All Materials</a>
13           <a href="/Discussions" class="btn">View Discussions</a>
14           <a href="/MyMaterials" class="btn">View Status of My Materials</a>
15           <a href="/EditUploads" class="btn">Edit My Uploads</a>
16         </nav>
17
18       <main class="main-content">
19
20         </div>
21         <div class="front-button">
22           <nav class="quick-links">
23             <div class="upload-dropdown">
24               <button onclick="toggleUploadMenu()" class="btn">Upload</button>
25               <div id="uploadMenu">
26                 <a href="/Upload">Upload Material</a>
27                 <a href="/Create">Create Post</a>
28               </div>
29             </div>
30           </nav>
31           <script>
32             function toggleUploadMenu() {
33               var menu = document.getElementById('uploadMenu');
34               menu.style.display = (menu.style.display === 'block') ? 'none' : 'block';
35               // Hide menu when clicking outside
36               document.addEventListener('click', function handler(e) {
37                 if (!menu.contains(e.target) && e.target.className !== 'btn') {
38                   menu.style.display = 'none';
39                   document.removeEventListener('click', handler);
40                 }
41               });
42             }
43           </script>
44         </div>
45       </main>
46     </body>
47   </html>

```

HomeView.hbs

```

view > LoginView.hbs > html > body > div.auth-container
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>ERMAD - Login</title>
7       <link rel="stylesheet" href="/style.css">
8     </head>
9     <body>
10    <div class="auth-container">
11      <div class="Main-img">
12        
13      </div>
14      <div class="auth-form">
15        <h1>Login</h1>
16        <form id="login-form" method="POST" action="/login">
17          <div class="form-group">
18            <label for="username">Username</label>
19            <input type="text" id="username" name="username" required>
20          </div>
21          <div class="form-group">
22            <label for="password">Password</label>
23            <input type="password" id="password" name="password" required>
24          </div>
25          <button type="submit" class="btn">LOGIN</button>
26        </form>
27        <p class="auth-link">Don't have an account? <a href="/register">Register as new user</a></p>
28      </div>
29    </div>
30    <script src="../control/main.js"></script>
31  </body>
32 </html>
33
34

```

LoginView.hbs

```

19   <h1>Manage Files</h1>
20   <div class="back-button-container">
21     <button onclick="window.location.href='/Admin'" class="back-button">
22       <img class="arrow-icon" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-miterlimit="10" style="vertical-align: middle;"/>
23     </button>
24   </div>
25   <form method="GET" action="/FileLists" class="sort-form">
26     <label for="sort-select">Sort by:</label>
27     <select id="sort-select" name="sort" onchange="this.form.submit()">
28       <option value="date" {{#if (eq sort 'date')}}selected{{/if}}>Date</option>
29       <option value="category" {{#if (eq sort 'category')}}selected{{/if}}>Category</option>
30       <option value="alphabetic" {{#if (eq sort 'alphabetic')}}selected{{/if}}>Alphabetic</option>
31     </select>
32   </form>
33   <main class="main-content">
34     <table>
35       <thead>
36         <tr>
37           <th>ID</th>
38           <th>Title</th>
39           <th>Uploaded by</th>
40           <th>Description</th>
41           <th>Category</th>
42           <th>Date</th>
43           <th>Status</th>
44           <th>Actions</th>
45         </tr>
46       </thead>
47     </table>
48

```

ManageFileView.hbs (1/2)

```

49   <tbody>
50     {{#each files}}
51       <tr>
52         <td>{{inc @index}}</td>
53         <td>{{this.title}}</td>
54         <td>{{this.uploadedBy}}</td>
55         <td>{{this.description}}</td>
56         <td>{{this.Category}}</td>
57         <td>{{formatDate this.date "DD/MM/YYYY"}}</td>
58         <td>{{this.status}}</td>
59         <td>
60           <form action="/filesLists/delete/{{this._id}}" method="POST" onsubmit="return confirmDelete();">
61             <button type="submit" class="delete-btn">Delete</button>
62             <a href="/uploads/{{this.file}}" target="_blank" class="btn">Open</a>
63           </form>
64         </td>
65       {{/each}}
66     </tbody>
67   </table>
68   <script>
69     function confirmDelete() {
70       return confirm('Are you sure you want to delete this user?');
71     }
72   </script>
73   <script src="../control/main.js"></script>
74 </body>
75 </html>
76
77

```

ManageFileView.hbs (2/2)

```

view > ManagePostView.hbs > html > head > title
  2   <html lang="en">
  3     <body>
  4       <form method="GET" action="/PostsLists" class="sort-form">
  5         <label for="sort-select">Sort by:</label>
  6         <select id="sort-select" name="sort" onchange="this.form.submit()">
  7           <option value="date" {{#if (eq sort 'date')}}selected{{/if}}>Date</option>
  8           <option value="category" {{#if (eq sort 'category')}}selected{{/if}}>Category</option>
  9           <option value="alphabetic" {{#if (eq sort 'alphabetic')}}selected{{/if}}>Alphabetic</option>
 10         </select>
 11       </form>
 12       <main class="main-content">
 13         <table>
 14           <thead>
 15             <tr>
 16               <th>ID</th>
 17               <th>Title</th>
 18               <th>Description</th>
 19               <th>Category</th>
 20               <th>Date</th>
 21               <th>Uploaded By</th>
 22               <th>Actions</th>
 23             </tr>
 24           </thead>
 25           <tbody>
 26             {{#each posts}}
 27               <tr>
 28                 <td>{{inc @index}}</td>
 29                 <td>{{this.Title}}</td>
 30                 <td>{{this.Description}}</td>
 31                 <td>{{this.Category}}</td>
 32                 <td>{{formatDate this.UploadDate "DD/MM/YYYY"}}</td>
 33                 <td>{{this.uploadedBy}}</td>
 34               </tr>
 35             {{/each}}
 36           </tbody>
 37         </table>
 38       </main>
 39       <script>
 40         function confirmDelete() {
 41           return confirm('Are you sure you want to delete this user?');
 42         }
 43       </script>
 44       <script src="/control/main.js"></script>
 45     </body>
 46   </html>

```

ManagePostView.hbs (1/2)

```

57       <td>
58         <form action="/Posts/delete/{{this._id}}" method="POST" onsubmit="return confirmDelete();">
59           <button type="submit" class="delete-btn">Delete</button>
60         </form>
61       </td>
62     </tr>
63   {{/each}}
64 </tbody>
65 </main>
66 <script>
67   function confirmDelete() {
68     return confirm('Are you sure you want to delete this user?');
69   }
70 </script>
71 <script src="/control/main.js"></script>
72 </body>
73 </html>

```

ManagePostView.hbs (2/2)

```

16   <h1>Manage Users</h1>
17   <div class="back-button-container">
18     <button onclick="window.location.href='/Admin'" class="back-button">
19       <img alt="arrow icon" data-bbox="338 538 488 568" />
20     </button>
21   </div>
22   <main class="main-content">
23     <table>
24       <thead>
25         <tr>
26           <th>ID</th>
27           <th>Username</th>
28           <th>Email</th>
29           <th>Phone</th>
30           <th>Actions</th>
31         </tr>
32       </thead>
33       <tbody>
34         {{#each users}}
35           <tr>
36             <td>{{inc @index}}</td>
37             <td>{{this.username}}</td>
38             <td>{{this.email}}</td>
39             <td>{{this.phone}}</td>
40             <td>
41               <form action="/Userlists/delete/{{this._id}}" method="POST" onsubmit="return confirmDelete();">
42                 <button type="submit" class="delete-btn">Delete</button>
43               </form>
44             </td>
45           </tr>
46         {{/each}}
47       </tbody>
48     </table>

```

ManageUserView.hbs

```

31   <div class="back-button-container">
32     <button onclick="window.location.href='/home'" class="back-button">
33       <svg class="arrow-icon" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2">
34         <path d="M5 12h14M12 5l7 7-7 7"/>
35       </svg>
36     </button>
37   </div>
38
39   <div class="front-button">
40     <nav class="quick-links">
41       <div class="upload-dropdown">
42         <button onclick="toggleUploadMenu()" class="btn">Upload</button>
43         <div id="uploadMenu">
44           <a href="/Upload">Upload Material</a>
45           <a href="/Create">Create Post</a>
46         </div>
47       </div>
48     </nav>
49   </div>
50
51   <script>
52     function toggleUploadMenu() {
53       var menu = document.getElementById('uploadMenu');
54       menu.style.display = (menu.style.display === 'block') ? 'none' : 'block';
55       // Hide menu when clicking outside
56       document.addEventListener('click', function handler(e) {
57         if (!menu.contains(e.target) && e.target.className !== 'btn') {
58           menu.style.display = 'none';
59         }
60       });
61     }
62   </script>

```

MyMaterialsView.hbs (1/2)

```

54   <main>
55     <table>
56       <tr>
57         <thead>
58           <th>Title</th>
59           <th>Status</th>
60         </thead>
61
62       </tr>
63       {{#each files}}
64         <tr>
65           <td>{{this.title}}</td>
66           <td>
67             {{#if (eq this.status "verifying")}}Verifying{{/if}}
68             {{#if (eq this.status "accepted")}}Accepted{{/if}}
69             {{#if (eq this.status "declined")}}Declined{{/if}}
70           </td>
71           <td>
72             {{#if (eq this.status "accepted")}}
73               <a href="/uploads/{{this.file}}">View</a>
74             {{else}}
75               <span>Not available</span>
76             {{/if}}
77           </td>
78         </tr>
79       {{/each}}
80     </table>
81
82   </main>
83
84   <script src="../control/main.js"></script>
85
86 </body>
87 </html>

```

MyMaterialsView.hbs (2/2)

```

32   <div class="back-button-container">
33     <button onclick="window.location.href='/Discussions'" class="back-button">
34       <svg class="arrow-icon" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2">
35         <path d="M5 12h14M12 5l7 7-7 7"/>
36       </svg>
37     </button>
38   </div>
39   <main class="main-content">
40     <section class="post-details">
41       <article class="post">
42         <h2>{{post.Title}}</h2>
43         <p>{{post.Content}}</p>
44         <div class="file-meta">
45           <span class="post-author">Posted by: {{post_uploaded_by}}</span>
46           <p><span>Category : {{post.Category}}</span></p>
47           <p><span>post-date</span> Date :{{formatDate post.uploadDate "DD/MM/YYYY"}}</span></p>
48         </div>
49       </article>
50     </section>
51   </main>
52

```

PostDetailsView.hbs (1/2)

```

55   </main>
56   <section class="replies">
57     <h3>Replies:</h3>
58     {{#if replies.length}}
59       {{#each replies}}
60         <div class="reply">
61           <p class="reply-content">{{this.Reply}}</p>
62           <div class="reply-meta">
63             <span class="reply-author">{{this.ReplyBy}}</span>
64             <span class="reply-date">{{formatDate this.replyDate "DD/MM/YYYY"}}</span>
65           </div>
66         </div>
67       {{/each}}
68     {{else}}
69       <p>No replies yet.</p>
70     {{/if}}
71     <form class="reply-form" action="/reply/{{post._id}}" method="POST">
72       <textarea id="Reply" name="Reply" placeholder="Write your reply here..." required></textarea>
73       <button type="submit">Post Reply</button>
74     </form>
75   </section>
76
77   <script src="../control/main.js"></script>
78
79 </body>
80 </html>

```

PostDetailsView.hbs (2/2)

```

view > RegisterView.hbs > html > head > title
  2   <html lang="en">
  3   <body>
  4     <div class="auth-container">
  5       <div class="auth-form">
  6         <h1>Register</h1>
  7         <form id="register-form" method="POST" action="/register">
  8           <div class="form-group">
  9             <label for="reg-username">Username</label>
 10             <input type="text" id="reg-username" name="username" required>
 11           </div>
 12           <div class="form-group">
 13             <label for="reg-password">Password</label>
 14             <input type="password" id="reg-password" name="password" required>
 15           </div>
 16           <div class="form-group">
 17             <label for="reg-confirm-password">Confirm Password</label>
 18             <input type="password" id="reg-confirm-password" name="confirm-password" required>
 19           </div>
 20           <div class="form-group">
 21             <label for="email">Email</label>
 22             <input type="email" id="email" name="email" required>
 23           </div>
 24           <div class="form-group">
 25             <label for="phone">Phone Number</label>
 26             <input type="tel" id="phone" name="phone" required>
 27           </div>
 28           <div class="checkbox-label" style="display: flex; align-items: center;">
 29             <input type="checkbox" id="copyright-agreement" name="copyright-agreement" required style="margin-right: 10px;" checked="checked">
 30             <span>I hereby acknowledge and agree that I will not post, upload, or share any materials that
 31             violate the terms of service or user agreement.</span>
 32           </div>
 33           <div>
 34             <button type="submit" class="btn">REGISTER</button>
 35           </div>
 36         </form>
 37       </div>
 38     </div>
 39   </div>
 40 </body>
 41 </html>

```

RegisterView.hbs

```

31   <div class="back-button-container">
32     <button onclick="window.location.href='/home'" class="back-button">
33       <img alt="arrow icon" class="arrow-icon" href="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="none" stroke="currentColor" style="width: 1em; height: 1em; vertical-align: middle;"/>
34     </button>
35   </div>
36
37   <div class="front-button">
38     <nav class="quick-links">
39       <div class="upload-dropdown">
40         <button onclick="toggleUploadMenu()" class="btn">Upload</button>
41         <div id="uploadMenu">
42           <a href="/Upload">Upload Material</a>
43           <a href="/Create">Create Post</a>
44         </div>
45       </div>
46     </nav>
47     <script>
48       function toggleUploadMenu() {
49         var menu = document.getElementById('uploadMenu');
50         menu.style.display = (menu.style.display === 'block') ? 'none' : 'block';
51         // Hide menu when clicking outside
52         document.addEventListener('click', function handler(e) {
53           if (!menu.contains(e.target) && e.target.className !== 'btn') {
54             menu.style.display = 'none';
55             document.removeEventListener('click', handler);
56           }
57         });
58       }
59     </script>
60   </div>
61 
```

SearchResultView.hbs (1/2)

```

63   <main>
64     <section class="discussion-list">
65       <h2>Search Results{{#if query}} for "{{query}}{{/if}}</h2>
66       <h3>Posts:</h3>
67       {{#if posts.length}}
68         <ul>
69           {{#each posts}}
70             <li class="discussion-item">
71               <a href="/view/{{this._id}}" class="discussion-link">{{this.Title}}</a> {{this.Category}}
72             </li>
73           {{/each}}
74         </ul>
75       {{else}}
76         <p>No posts found.</p>
77       {{/if}}
78
79       <h3>Materials:</h3>
80       {{#if materials.length}}
81         <ul>
82           {{#each materials}}
83             <li class="discussion-item">
84               {{this.title}} {{this.Category}}
85               {{#if this_file}}
86                 <a href="/viewFile/{{this._id}}" class="discussion-link">{{this.title}}</a> {{this.Category}}
87               {{/if}}
88             </li>
89           {{/each}}
90         </ul>
91       {{else}}
92         <p>No materials found.</p>
93       {{/if}}
94     </section>
95   </main>

```

SearchResultView.hbs (2/2)

```

view > UploadView.hbs > ...
2  <html lang="en">
9   <body>
10    <main class="main-content">
11      <section class="upload-section">
12          <h2>Upload Course Material</h2>
13          <form id="upload-form" action="/upload" method="POST" enctype="multipart/form-data" onsubmit="return PostForm(this);">
14              <div class="form-group">
15                  <label for="file">Upload the file here</label>
16                  <input type="file" name="file" accept=".pdf,.zip" required>
17              </div>
18              <div class="form-group">
19                  <label for="title">Name</label>
20                  <input type="text" id="title" name="title" required>
21              </div>
22              <div class="form-group">
23                  <label for="description">Description</label>
24                  <textarea id="description" name="description" required></textarea>
25              </div>
26              <div class="form-group">
27                  <label for="Category">Category</label>
28                  <select id="Category" name="Category" required>
29                      <{#if categories}>
30                          <{#each categories}>
31                              <option >{this.Categoryname}</option>
32                          <{/each}>
33                      <{else}>
34                          <option disabled>No categories available</option>
35                      <{/if}>
36                  </select>
37              </div>
38              <button type="submit" class="btn">Upload</button>
39          </form>
40      </section>
41  </main>
42 </body>
43 </html>

```

UploadView.hbs

```

view > Verification.hbs > html > head > title
2  <html lang="en">
9   <body>
10    <form method="GET" action="/Verifications" class="sort-form">
11        <label form="sort-select">Sort by:</label>
12        <select id="sort-select" name="sort" onchange="this.form.submit()">
13            <option value="date" >{#if (eq sort 'date')}>selected{!/if}>Date</option>
14            <option value="category" >{#if (eq sort 'category')}>selected{!/if}>Category</option>
15            <option value="alphabetic" >{#if (eq sort 'alphabetic')}>selected{!/if}>Alphabetic</option>
16        </select>
17    </form>
18    <main class="main-content">
19        <table>
20            <thead>
21                <tr>
22                    <th>ID</th>
23                    <th>Title</th>
24                    <th>uploaded by</th>
25                    <th>Description</th>
26                    <th>date</th>
27                    <th>status</th>
28                    <th>Actions</th>
29                </tr>
30            </thead>
31            <tbody>
32                <{#each files}>
33                    <tr>
34                        <td>{inc @index}</td>
35                        <td>{this.title}</td>
36                        <td>{this.uploadedby}</td>
37                        <td>{this.description}</td>
38                        <td>{formatDate this.date "DD/MM/YYYY"}</td>
39                        <td>{this.status}</td>
40                    </tr>
41                </{#each}>
42            </tbody>
43        </table>
44    </main>
45 </body>
46 </html>

```

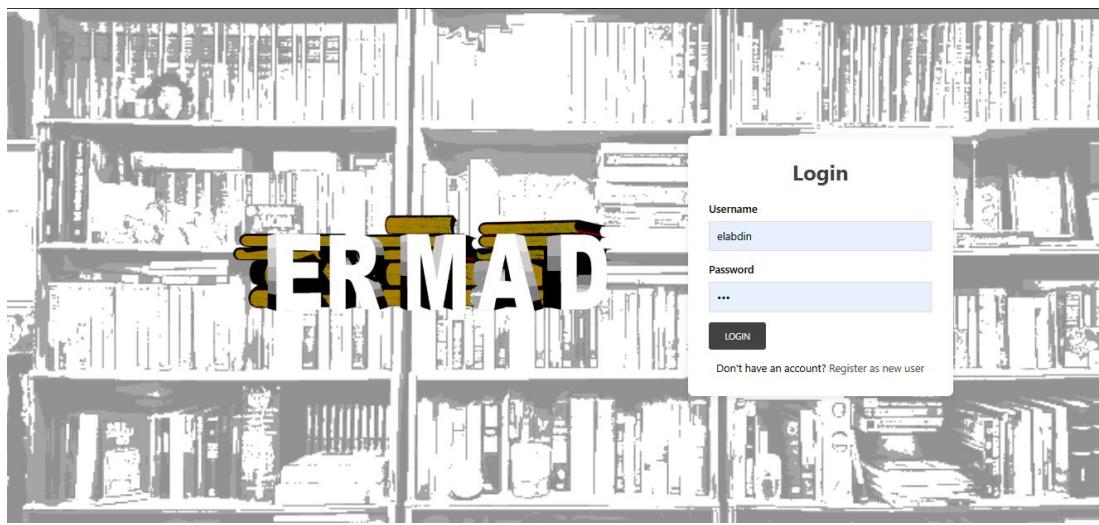
Verification.hbs (1/2)

```

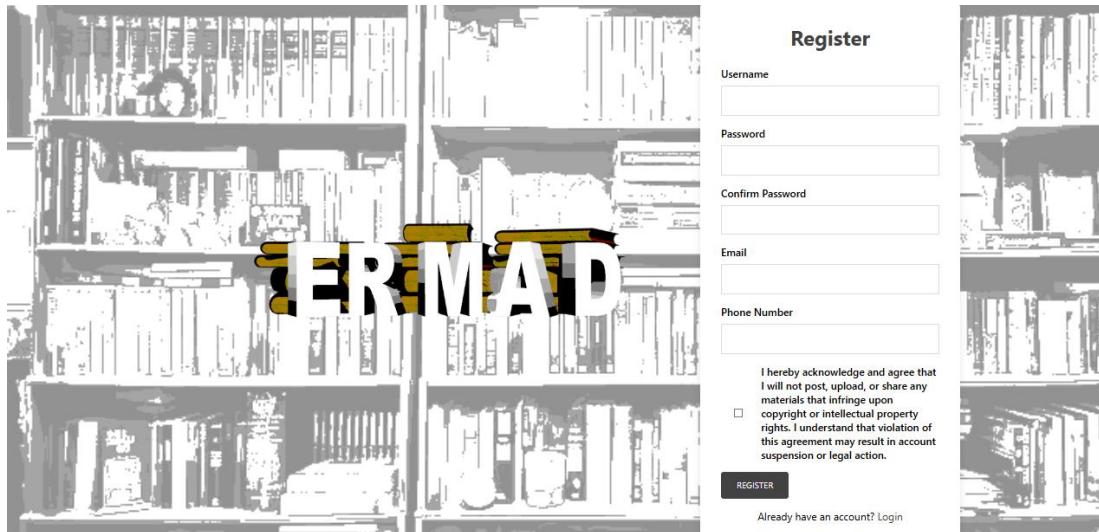
55   <form action="/Verifications/accept/{this._id}" method="POST" style="display:inline;">
56       <button type="submit" class="btn">Accept</button>
57   </form>
58   <form action="/Verifications/decline/{this._id}" method="POST" style="display:inline;">
59       <button type="submit" class="delete-btn">Decline</button>
60   </form>
61   <a href="/uploads/{this.file}" target="_blank" class="btn">View</a>
62   </td>
63   <{#each}>
64   </tbody>
65 </table>
66 </main>
67 <script>
68     function confirmDelete() {
69         return confirm('Are you sure you want to delete this user?');
70     }
71 </script>
72 <script src="../../control/main.js"></script>
73 </body>
74 </html>

```

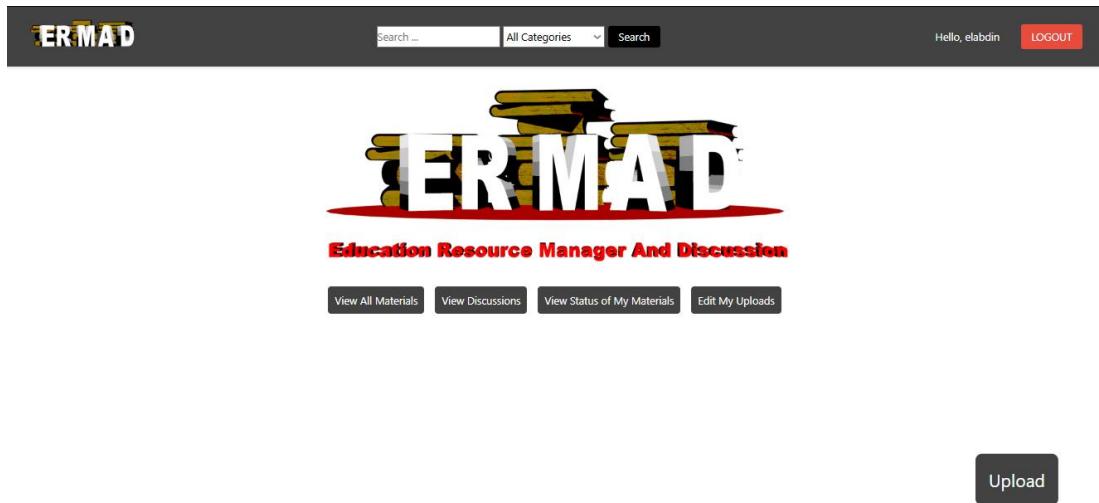
Verification.hbs (2/2)



Login Page



Register Page



Home Page



Upload Course Material

Upload the file here

No file chosen

Name

Description

Category

Upload Course Material Page



Create Post

Title

Description

Category

Create Post Page



Sort by: Category ▾

Biology

How do vaccines work in the body?
Can someone explain the biological process of how ...

Best way to memorize biological terms?
What are your tips for memorizing complex biologic...

Chemistry

Balancing chemical equations tips
What methods do you use to balance chemical equati...

Discussion View Page

TERMAD

Search ...

Hello, elabdin

How do vaccines work in the body?
 Can someone explain the biological process of how vaccines help us fight diseases?
Posted by: Betson10
Category: Biology
Date: 05/06/2025

Replies

Vaccines work by training your immune system to recognize and combat pathogens (like viruses or bacteria) without causing the actual disease.
elabdin 30/06/2025

Write your reply here...

Discussion details Page

TERMAD

Search ...

Files

Title	Description	Category	Actions
PROBABILITY AND STATISTICAL DATA ANALYSIS	PROBABILITY AND STATISTICAL DATA ANALYSIS	Computer Science	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Posts

Title	Description	Category	Actions
Computer Fundamentals	What are the main components of a computer system? Explain the difference between hardware and software. What is the role of the CPU in a computer? How does RAM differ from ROM? What are input and output devices? Give examples.	Computer Science	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Edit my uploads Page

TERMAD

Search ...

Edit File

Name

Description

Category

Edit Upload Page

TERMAD	<input type="text" value="Search ..."/>	All Categories	<input type="button" value="Search"/>	<input type="button" value="LOGOUT"/>
←				
Title	Status	Status		

PROBABILITY AND STATISTICAL DATA ANALYSIS Verifying Not available

View status Page

TERMAD	<input type="text" value="Search ..."/>	All Categories	<input type="button" value="Search"/>	Hello, elabdin	<input type="button" value="LOGOUT"/>
←				Sort by:	Category
Biology					
Understanding How Vaccines Work					
No preview available Diseases that vaccines prevent can be dangerous, o...					
Computer Science					
PROBABILITY AND STATISTICAL DATA ANALYSIS					
No preview available PROBABILITY AND STATISTICAL DATA ANALYSIS...					
					<input type="button" value="Upload"/>

View files Page

No preview available
Understanding How Vaccines Work

Diseases that vaccines prevent can be dangerous, or even deadly. Vaccines greatly reduce the risk of infection by working with the body's natural defenses to safely develop immunity to disease. This fact sheet explains how the body fights infection and how vaccines work to protect people by producing immunity.

Uploaded by: elabdin
Date:

File View and download Page



Search posts...
All Categories
Search

Hello, elabdin
LOGOUT



Search Results for "best"

Posts

- Best programming language for beginners? (Computer Science)
- Best way to memorize biological terms? (Biology)
- How to remember historical dates? (History)

Materials

No materials found.

[Upload](#)

Search result Page



Logout



Education Resource Manager And Discussion

[Manage Users](#) [Manage Posts](#) [Manage Files](#) [Verify files](#) [Manage Categories](#)

Admin home Page



Logout

Manage Users

[Back](#)

ID	Username	Email	Phone	Actions
1	admin	altyusufzain@gmail.com	1169721623	Delete
2	John	altyusufzain@gmail.com	1169721623	Delete
3	elabdin	altyusufzain@gmail.com	97433078987	Delete
4	Ahmed	zainelsaikhy@gmail.com	97433078987	Delete
5	bob	zainelsaikhy@gmail.com	334	Delete
6	Alpher	alpha@mail.com	123456789	Delete
7	Betson10	Beta@mail.com	987654321	Delete
8	charlie2025	charlie@mail.com	112233445	Delete

Manage Users Page

TER MAD

Manage Posts

[Logout](#)

Sort by: Date

ID	Title	Description	Category	Date	Uploaded By	Actions
1	Favorite science experiment in school?	Share your favorite science experiment you did in school and what you learned from it.	Science	02/06/2025	Ahmed	Delete
2	Tips for solving calculus problems	What strategies do you use to solve difficult calculus problems efficiently?	Mathematics	03/06/2025	bob	Delete
3	Why is math important in everyday life?	Discuss how mathematics is used in daily activities and why it's a crucial skill.	Mathematics	04/06/2025	Alpher	Delete
4	How do vaccines work in the body?	Can someone explain the biological process of how vaccines help us fight diseases?	Biology	05/06/2025	Betson10	Delete
5	Best way to memorize biological terms?	What are your tips for memorizing complex biological vocabulary and processes?	Biology	06/06/2025	charlie2025	Delete
6	Explaining Newton's Laws of Motion	Can someone provide simple examples to explain Newton's three laws of motion?	Physics	07/06/2025	Gaster	Delete

Manage Posts Page

TER MAD

Manage Files

[Logout](#)

Sort by: Date

ID	Title	uploaded by	Description	Category	date	status	Actions
1	Understanding How Vaccines Work	elabdin	Diseases that vaccines prevent can be dangerous, or even deadly. Vaccines greatly reduce the risk of infection by working with the body's natural defenses to safely develop immunity to disease. This fact sheet explains how the body fights infection and how vaccines work to protect people by producing immunity.	Biology	30/06/2025	accepted	Delete Open
2	PROBABILITY AND STATISTICAL DATA ANALYSIS	elabdin	PROBABILITY AND STATISTICAL DATA ANALYSIS	Computer Science	30/06/2025	accepted	Delete Open

Manage Files Page

TER MAD

Verify Files

[Logout](#)

Sort by: Date

ID	Title	uploaded by	Description	date	status	Actions
1	Understanding How Vaccines Work	elabdin	Diseases that vaccines prevent can be dangerous, or even deadly. Vaccines greatly reduce the risk of infection by working with the body's natural defenses to safely develop immunity to disease. This fact sheet explains how the body fights infection and how vaccines work to protect people by producing immunity.	30/06/2025	verifying	Accept Decline View
2	PROBABILITY AND STATISTICAL DATA ANALYSIS	elabdin	PROBABILITY AND STATISTICAL DATA ANALYSIS	30/06/2025	verifying	Accept Decline View

Verify Files Page

Manage Categories

Category Name: **Add Category**

ID	Category Name	Number of Materials	Actions
1	Biology		Delete
2	Biomedical		Delete
3	Chemistry		Delete
4	Computer Science		Delete
5	Engineering		Delete
6	History		Delete
7	Literature		Delete
8	Mathematics		Delete

Manage Categories Page