

CPSC 332 Web Project Deliverable 4

Project Management Application

Evan Shoemaker

12/8/2022

GitHub URL

<https://github.com/GUWebDevelopment/cpsc-332-web-development-final-project-Colexeco.git>

Functional Requirement 1

Users must be able to create project/task entries (front)

User enters title, description, deadline information and task information as desired...

The screenshot shows a web form titled "Create New Project" in green text on a black background. The form contains two identical sections. Each section has the following elements: a green label "Title:" followed by a white text input field with the placeholder "add a title here"; a green label "Description:" followed by a white text input field with the placeholder "add a description here"; a green label "Deadline:" followed by a white text input field with the placeholder "mm/dd/yyyy" and a small calendar icon; and a green label "Completed?" followed by a white checkbox. Below the first section are two green buttons: "Add task" and "Remove task". Below the second section is a single green button labeled "Create".

Functional Requirement 2


Users must be able to retrieve previously created project/task entries (back)

User clicks on the “All projects...” button...

Create New Project

Title:

Description:

Deadline: 

Completed? ☐

Add task

Remove task

Create

All projects...

Log Out

User is provided with central page to view all projects...

Project created successfully

View all projects here...

title	description	deadline	completed	tasks		
proj1	desc	Fri Dec 09 2022 16:00:00 GMT-0800 (Pacific Standard Time)	false	2	<div>View Tasks</div>	<div>Edit</div> <div>Delete</div>
f	f	Thu Dec 08 2022 16:00:00 GMT-0800 (Pacific Standard Time)	true	1	<div>View Tasks</div>	<div>Edit</div> <div>Delete</div>
Notasks	desc	Thu Dec 08 2022 16:00:00 GMT-0800 (Pacific Standard Time)	true	0		<div>Edit</div> <div>Delete</div>

Create a new project

Log Out

```

//READ
✓ app.get("/viewProjects", (req, res) => {
  ✓
  ✓   if (req.session.userId) {
  ✓       validateSession(req.session.userId, res);
  ✓
  ✓       projectResult.find(
  ✓           {},
  ✓           (err, results) => {
  ✓               console.log("Results: " + results)
  ✓               res.render("viewProjects.ejs", {
  ✓                   projectResults: results,
  ✓               });
  ✓           });
  ✓   } else { //not logged in
  ✓       return res.redirect("/login");
  ✓   }
  ✓ });

```

```

<body class="page-wrap">
  <h1>Project created successfully</h1>
  <table class="table">
    <caption>View all projects here...</caption>
    <tr>
      <th>title</th>
      <th>description</th>
      <th>deadline</th>
      <th>completed</th>
      <th>tasks</th>
    </tr>
    <% projectResults.forEach(function(response) { //debugger; %>
      <% console.log(response) %>
      <tr>
        <td id="title" class="table">
          <%= response.title %>
        </td>
        <td id="description" class="table">
          <%= response.description %>
        </td>
        <td id="deadline" class="table">
          <%= response.deadline %>
        </td>
        <td id="completed" class="table">
          <%= response.completed %>
        </td>
        <td id="tasks" class="table">
          <%= response.numtasks %>
          <% if (response.numtasks > 0) { %>
            <a href="/viewTasks/<%= response._id %>" class="button">View Tasks</a>
          <% } %>
        </td>
        <td class="table">
          <a href="/edit/<%= response._id %>" class="button">Edit</a>
          <a href="/delete/<%= response._id %>" class="button">Delete</a>
        </td>
      </tr>
    <% }); %>
  </table>
  <div class="page-wrap">
    <button class="button"><a class="button" href="/newProject">Create a new project</a></button>
    <button class="button"><a class="button" href="/logout">Log Out</a></button>
  </div>
</body>

```

Functional Requirement 3

Users must be able to modify projects/tasks to reflect progress (front)

User clicks “edit” on project they would like to modify...

Project created successfully

View all projects here...

title	description	deadline	completed	tasks			
proj1	desc	Fri Dec 09 2022 16:00:00 GMT-0800 (Pacific Standard Time)	false	2	View Tasks	Edit	Delete
f	f	Thu Dec 08 2022 16:00:00 GMT-0800 (Pacific Standard Time)	true	1	View Tasks	Edit	Delete
Notasks	desc	Thu Dec 08 2022 16:00:00 GMT-0800 (Pacific Standard Time)	true	0		Edit	Delete


[Create a new project](#) [Log Out](#)

User edits desired fields and clicks “Edit (Save Changes)...”

Edit Project

Title

Description

Deadline 

Completed

[All Projects \(No changes saved\)](#) [Edit Tasks \(No Changes Saved\)](#) [Edit \(Save Changes\)](#)

Functional Requirement 4

Users must be able to retrieve past modifications to view progress (back)

Changes are sent to database when user clicks “Edit (Save Changes)...”

User can view project with new changes...

Project created successfully

[View all projects here...](#)

title	description	deadline	completed	tasks		
proj1	desc	Fri Dec 09 2022 16:00:00 GMT-0800 (Pacific Standard Time)	false	2	View Tasks	Edit Delete
f	f	Thu Dec 08 2022 16:00:00 GMT-0800 (Pacific Standard Time)	true	1	View Tasks	Edit Delete
Notasks	desc	Thu Dec 08 2022 16:00:00 GMT-0800 (Pacific Standard Time)	true	0	View Tasks	Edit Delete

[Create a new project](#)

[Log Out](#)

```
//READ
✓ app.get("/viewProjects", (req, res) => {
  ✓   if (req.session.userId) {
  ✓     validateSession(req.session.userId, res);
  ✓
  ✓     projectResult.find(
  ✓       {},
  ✓       (err, results) => {
  ✓         console.log("Results: " + results)
  ✓         res.render("viewProjects.ejs", {
  ✓           projectResults: results,
  ✓         });
  ✓       });
  ✓   } else { //not logged in
  ✓     return res.redirect("/login");
  ✓   }
  });
```

```

//UPDATE
app.route("/edit/:id")
  .get((req, res) => {
    if (req.session.userId) {
      validateSession(req.session.userId, res);

      let id = req.params.id;

      id = sanitize(id); //sanitization

      projectResult.findById(
        id,
        (err, results) => {
          if (err) {
            return res.render("error.ejs", {
              errors: err
            });
          }

          console.log("Found project! ");
          console.log(results);

          let result = {
            _id: id,
            title: results.title,
            description: results.description,
            deadline: results.deadline,
            completed: results.completed,
            tasks: results.tasks,
          };

          res.render("editProject.ejs", {
            response: result
          });
        }
      );
    } else { //not logged in
      return res.redirect("/login");
    }
  })
})

```

```

},
.post(function (req, res) {
  if (req.session.userId) {
    validateSession(req.session, res);

    let id = req.params.id;

    id = sanitize(id); //sanitization

    let title = req.body.title;
    let description = req.body.description;
    let deadline = req.body.deadline;
    let completed = req.body.completed;
    let tasks = req.body.tasks;

    //create an object it's possible to validate on
    let data = {
      title: title,
      description: description,
      deadline: deadline,
      completed: completed,
      tasks: tasks
    }

    //validate
    let validation = new Validator(data, projectRules,
    {
      required: "A valid :attribute is required",
      boolean: ":attribute must be true/false",
      date: "Please provide a valid date",
    });

    console.log("Validation Passes: " + validation.passes() + " Validation Fails: " + validation.fails());

    if (validation.fails()) {
      let errorslist = {
        title: validation.errors.first("title"),
        description: validation.errors.first("description"),
        deadline: validation.errors.first("deadline"),
        completed: validation.errors.first("completed")
      };
    }
  }
});

```

```

    if (validation.fails()) {
      let errorsList = {
        title: validation.errors.first("title"),
        description: validation.errors.first("description"),
        deadline: validation.errors.first("deadline"),
        completed: validation.errors.first("completed")
      };

      res.render("error.ejs", {
        errors: 3,
        errors: errorsList,
      });
    } else {

      projectResult
        .where({_id: id })
        .updateOne({
          $set: {
            title: title,
            description: description,
            deadline: deadline,
            completed: completed,
            tasks: tasks,
          }
        })
        .exec(function (err, result) {
          if (err) return res.send(err);
          res.redirect("/viewProjects");
          console.log(`Successfully updated ${result.modifiedCount} record`);
        });
    }
  } else { //not logged in
    return res.redirect("/login");
  }
});

```

Functional Requirement 5

Users must be able to add deadlines to projects/tasks (front)

User enters deadline for project and any desired tasks...

Create New Project

Title:

Description:

Deadline: 

Completed? ☐

title:

description:

deadline: 

Completed? ☐

Functional Requirement 6

Users must be able to retrieve previously created deadlines for projects/tasks (back)

User can view deadlines at the central project page...

Project created successfully

[View all projects here...](#)

title	description	deadline	completed	tasks		
proj1	desc	Fri Dec 09 2022 16:00:00 GMT-0800 (Pacific Standard Time)	false	2	<input type="button" value="View Tasks"/>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
f	f	Thu Dec 08 2022 16:00:00 GMT-0800 (Pacific Standard Time)	true	1	<input type="button" value="View Tasks"/>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Notasks	desc	Thu Dec 08 2022 16:00:00 GMT-0800 (Pacific Standard Time)	true	0		<input type="button" value="Edit"/> <input type="button" value="Delete"/>

```

//READ
app.get("/viewProjects", (req, res) => {
  if (req.session.userId) {
    validateSession(req.session.userId, res);

    projectResult.find(
      {},
      (err, results) => {
        console.log("Results: " + results)
        res.render("viewProjects.ejs", {
          projectResults: results,
        });
      });
  } else { //not logged in
    return res.redirect("/login");
  }
});

```

Functional Requirement 7

Users must be able to fill out descriptions for projects/tasks (front)

User enters description(s) for project and any required tasks...

Create New Project

Title:

Description:

Deadline: 

Completed? ☐

Add task

Remove task

title:

description:

deadline: 

Completed? ☐

Create

Functional Requirement 8

Users must be able to retrieve previously created descriptions for projects/tasks (back)

User navigates to view all projects...

Project created successfully

View all projects here...

title	description	deadline	completed	tasks			
proj1	desc	Fri Dec 09 2022 16:00:00 GMT-0800 (Pacific Standard Time)	false	2	View Tasks	Edit	Delete
f	f	Thu Dec 08 2022 16:00:00 GMT-0800 (Pacific Standard Time)	true	1	View Tasks	Edit	Delete
Notasks	desc	Thu Dec 08 2022 16:00:00 GMT-0800 (Pacific Standard Time)	true	0		Edit	Delete

Create a new projectLog Out

```
//READ
app.get("/viewProjects", (req, res) => {
  if (req.session.userId) {
    validateSession(req.session.userId, res);

    projectResult.find(
      {},
      (err, results) => {
        console.log("Results: " + results)
        res.render("viewProjects.ejs", {
          projectResults: results,
        });
      });
  } else { //not logged in
    return res.redirect("/login");
  }
});
```

Functional Requirement 9

Users must be able to log in and out of application using a username and password (front)

User logs in with username and password...

User creates a new account...

Create a New Account...

Username

Password

Password

User logs into app with their username and password...

Log In to App...

Username

Password

Haven't created an account yet?

Functional Requirement 10

Users must be able to log back in with previous name and password from database (back)

User enters credentials to log back in...

Log In to App...

Username

Password

Log In

Haven't created an account yet?

Sign Up

```

app.route("/signUp")
  .get((req, res) => {
    let errors = {
      usernameError: "",
    }
    res.render("signUp.ejs", errors);
  })
  .post((req, res) => {
    if (req.body.username && req.body.password && req.body.passwordConfirmation) {
      var userData = userCredentials({
        username: req.body.username,
        password: req.body.password,
      });

      userData.save(function (err, user) {
        if (err) {
          let errors = {
            usernameError: "Error: " + err
          }
          res.render("signUp.ejs", errors);
        } else {
          return res.redirect("/newProject");
        }
      });
    }
  })

app.route("/login")
  .get((req, res) => {
    let errors = {
      usernameError: ""
    }
    res.render("login.ejs", errors);
  })
  .post((req, res) => {
    if (req.body.username && req.body.password) {
      var userData = {
        username: req.body.username,
        password: req.body.password,
      }
      let temp = userCredentials.authenticate(userData, req, res);
    }
  });

```

