

# CPSC 332 Web Project Deliverable 3

## Project Management Application

Evan Shoemaker

11/30/2022

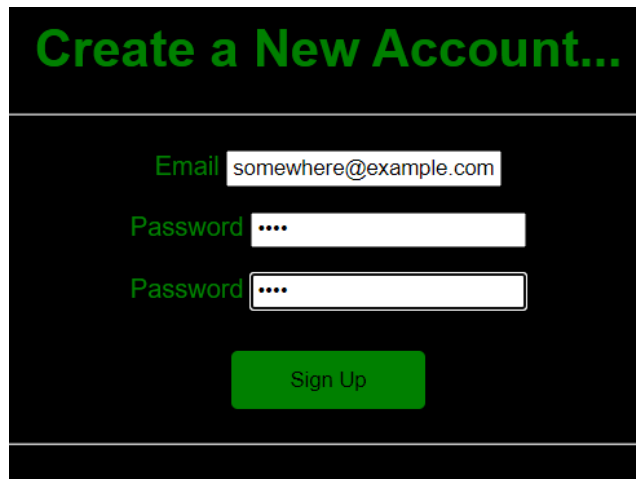
GitHub URL

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

### Functional Requirement 1

Users must be able log back in with a previously created username and password

User clicks “sign up” and creates a new account and signs in with a new account...

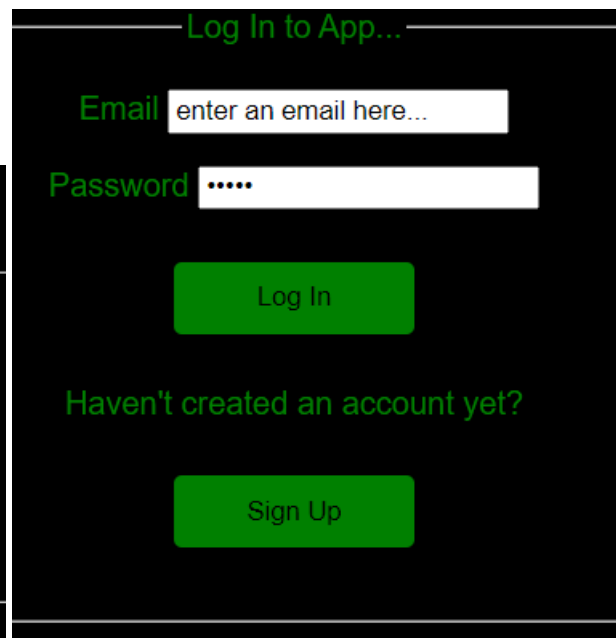


**Create a New Account...**

Email

Password

Password



**Log In to App...**

Email

Password

Haven't created an account yet?

```

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

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

```

User credentials are entered into the database

```

_id: ObjectId('6388452283f8b0b219f7a8b5')
email: "example@somewhere.com"
password: "$2b$10$mmbVEnhvjcV9GBnYuY0t0XejKEgez1gt8pespZHMZ.K2RgWulWbq"
__v: 0

```

## Functional Requirement 2

Users must be able to retrieve projects/task entries

User enters project attributes...

Create New Project

Title: invade mars

Description: stop at taco bell first though

Deadline: 11/11/1111

Completed? ☒

Add task

Remove task

title: establish taco bell on mars

description: asap

deadline: 11/11/1111

Completed? ☒

Create

All projects...

Log Out

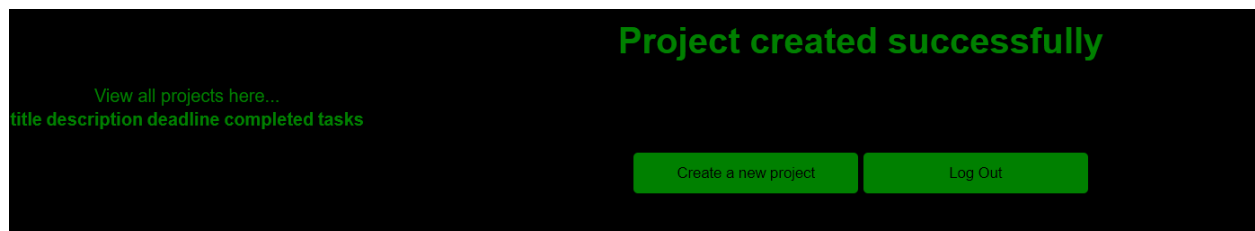
Project attributes are entered into database and displayed for retrieval...

```
//READ
app.get("/viewProjects", (req, res) => {
  if (req.session.userId) {
    validateSession(req.session.userId, res);
    let tempResult = {
      title: "tempTitle",
      description: "description",
      deadline: "11/11/1111",
      completed: true,
      tasks: [],
    }
    projectResult.find(
      {},
      (err, results) => {
        console.log(results)
        res.render("viewProjects.ejs", {
          projectResults: tempResult,
        });
      });
  } else { //not logged in
    return res.redirect("/login");
  }
});
```

## Functional Requirement 3

Users must be able to retrieve previously created modifications to view progress

User selected desired project to view...



Database call to retrieve desired project for viewing...

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

      let id = req.params.id;

      projectResult.findById(
        id,
        (err, results) => {
          console.log("Found result: ");
          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;
```

```

model('updateProject/123', {isOver1000})
.post(function (req, res) {
  if (req.session.userId) {
    validateSession(req.session, res);

    let id = req.params.id;

    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;

    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}`);
      });
  } else { //not logged in
    return res.redirect("/login");
  }
});
});

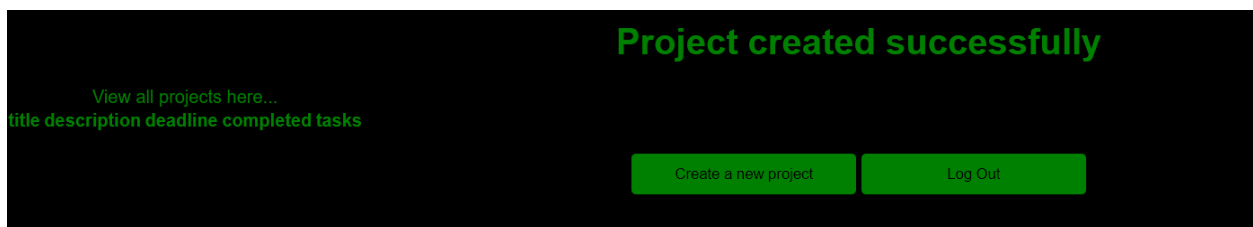
```

(continued from above)

## Functional Requirement 4

Users must be able to retrieve previously created deadlines to view progress

User chooses desired project to view...



Display all projects...

```

//READ
app.get("/viewProjects", (req, res) => {

  if (req.session.userId) {
    validateSession(req.session.userId, res);
    let tempResult = {
      title: "tempTitle",
      description: "description",
      deadline: "11/11/1111",
      completed: true,
      tasks: [],
    }

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

```

## Functional Requirement 5

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

User views all projects and selects which one to view...

```

//READ
app.get("/viewProjects", (req, res) => {

  if (req.session.userId) {
    validateSession(req.session.userId, res);
    let tempResult = {
      title: "tempTitle",
      description: "description",
      deadline: "11/11/1111",
      completed: true,
      tasks: [],
    }

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

```

Projects are stored in database...

```
_id: ObjectId('6388435583f8b0b219f7a893')  
completed: true  
> tasks: Array  
__v: 0
```

```
_id: ObjectId('638850f2e42ba85bf48a76de')  
completed: false  
> tasks: Array  
__v: 0
```