



Machine Project Specifications

All Sections



CCAPDEV

Specs due: **February 9, 2020 (Sun)**

Machine Project

Phase 1 due: **March 1, 2020 (Sun)**

AY1920 - Term 2

Phase 2 due: **March 22, 2020 (Sun)**

Phase 3 due: **April 12, 2020 (Sun)**


Instructions and Deliverables

Your task is to create a web application of your choice based on the minimum specifications required.

The project is **at most** done in **groups of 3**. A person cannot discuss or ask about design or implementation with other persons, with the exception of the teacher and his/her groupmates. Copying other people's work or working in collaboration with other teams is not allowed and is punishable by a grade of 0.0 for the entire CCAPDEV course. A discipline case may be filed with the Discipline Office. In short, do not risk it; the consequences are not worth the reward.

Below are the different requirements and corresponding submission dates for each.

Project Specifications Document

Since the group can choose their own topic, a specific project specification document will not be given out. Rather, the group is expected to write their own MP specifications. See [CCAPDEV_MP_Specs_Template.pdf](#)  for the sample templates. All text highlighted in yellow must be modified by the group.

Submission date: **February 9, 2020 (Sunday) 11:59 PM through Canvas**

MP Phase 1

For this phase, the group is required to build the application using HTML, CSS, and JavaScript. The group may use their preferred JavaScript libraries but ensure that this is compatible to run with Node.js. MP Phase 2 requires the application to be connected to a Node.js server. The group may also choose their preferred interface libraries like Bootstrap and Meteor.

The main requirement for this phase is that ***all views by all users should be visible and navigable from the first page***. Data may be hardcoded for now. There should be at least 5 dummy data present for each collection. Data encoded should be life-like (meaning no lorem ipsum generated paragraphs).

Submission date: **March 1, 2020 (Sunday) 11:59 PM through Canvas**

MP Phase 2

For this phase, you are required to submit a Node.js project that can be connected to locally by accessing localhost:3000. Architecture needs not to implement MVC yet but highly encouraged. For encouragement, MP Phase 3 is required to be in MVC.

Model

- Database design must be complete and the schema must be submitted.
- All database related files should be in the model folder.

View

- All views must be visible and can be navigated to from the first page.
- All forms must use the appropriate HTTP methods (i.e. Login forms use POST, search queries use GET, etc).
- All views are expected to be in the views folder. Convert files to handlebars/ejs/pug files as necessary.

Controller

- The backend is expected to be in NodeJS. Database dialect is encouraged to be MongoDB (Mongoose as an ODM is accepted). Using third party database servers such as mlab is allowed. The use of different server scripts and database is allowed, like the Node.js version of Firestore.

Sample requirement (routes are complete but backend logic has yet to be implemented)

```
app.get("/", (req, res) => {
  // user wants to access the main page

  // if not logged in
  res.render("index.hbs")
  // else,
  // res.render("home.hbs")
})

app.post("/login", urlencoder, (req, res) => {
  // user wants to login
  if (res.body.username === "admin" && res.body.password === "1234") {
    res.render("home.hbs")
  } else {
    res.render("index.hbs")
  }
})

app.get("/post", urlencoder, (req, res) => {
  // user wants to view a post
  var id = res.body.postid

  // To implement in phase 3: get post with id

  res.render("post.hbs")
})

app.post("/comment", urlencoder, (req, res) => {
  // user made a comment
  var comment = res.body.comment
  var pid = res.body.postid

  // To implement in phase 3: add to db

  res.render("post.hbs")
})

app.listen(3000);
```

Submission date: **March 22, 2020 (Sunday) 11:59 PM through Canvas**

MP Phase 3

For this phase, groups are expected to act upon the feedback given from MP Phase 2.

- The complete program must be completely functional with all the features running.
- The system must be deployed. Deploying to Heroku is recommended, but other options like

DigitalOcean is allowed. Azure and AWS have some DB hosting as well if mlab won't work for you. These have some free options for students, just use the dlsu.edu.ph email.

- The project must follow the model-view-controller (MVC) architecture.
- The content seen/manipulated in the views must be reflected in the database.
- There should at least be **20 dummy data** and **4 user accounts**.
- Sessions must be properly implemented.
- The system must check for cross-site scripting.
- All passwords should be hashed (use crypto) and database executions must account for injection attacks.
- All necessary CRUD operations for all collections are expected to be complete.
 - All database operations required in the final submission should already be implemented. Take note that it does not need to be "called" in the main controller yet.
- All NPM packages used for the project as well as third party libraries must be stated and indicated in the "About" page.

Submission date: **April 12, 2020 (Sunday) 11:59 PM through Canvas**

FAQs

The UI doesn't matter, right?

As much as possible, design your application to be user-friendly and pleasing to the user's eyes. Avoid shocking color combinations and ensure that fonts are readable. Aim for compatibility across browsers to ensure that the application will render correctly regardless of the user's browser.

Can we upload it to Heroku instead?

Yes! Do this if you already have bits of Node in your project already.

Can we use mlab?

Yes!

Can we use ejs/pug?

Yes!

Can we use Firebase in Web?

No, but you can use Firebase/Firestore in your Node.js.

Search entries or author

Unread



✓ **Subscribe**

↩ **Write a reply...**