

Web Application Development: Exam II

- This application (assignment checker) allows students to check their assignments and an administrator to add students' assignments.
- Users (students and an administrator) should be redirected to the login screen (**Figure 1**).
- When users log into the **non-administrative** account, they should be redirected to the page with their assignment(s) correctly displayed (**Figure 2**).
- When "admin" logs in, (s)he is redirected to the "admin" action (**Figure 3**). The administrative subsystem will allow administrators to add new assignments for students (**Figure 4**)

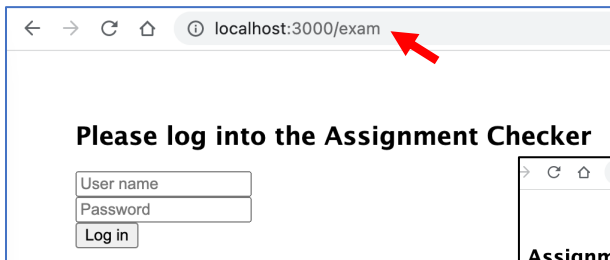
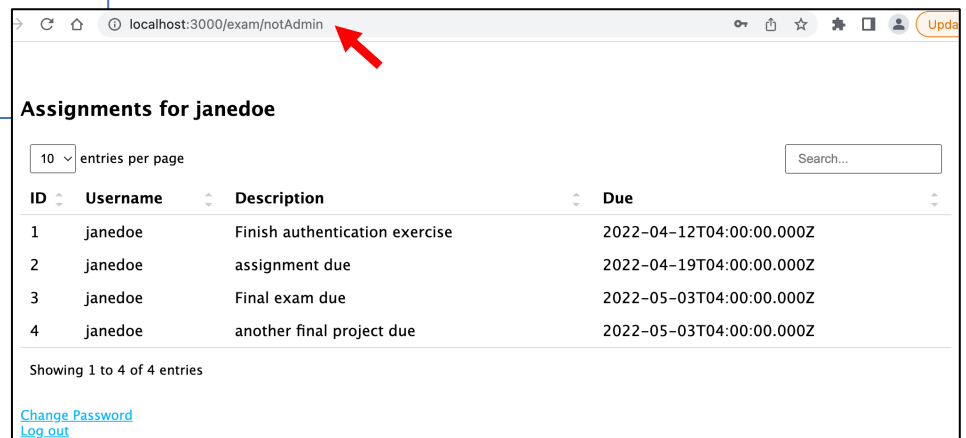


Figure 1



ID	Username	Description	Due
1	janedoe	Finish authentication exercise	2022-04-12T04:00:00.000Z
2	janedoe	assignment due	2022-04-19T04:00:00.000Z
3	janedoe	Final exam due	2022-05-03T04:00:00.000Z
4	janedoe	another final project due	2022-05-03T04:00:00.000Z

Figure 2

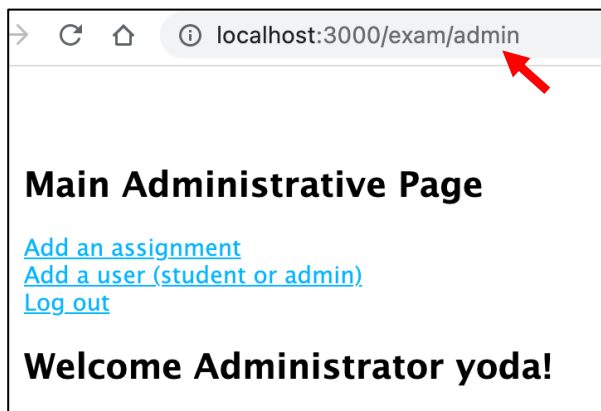


Figure 3

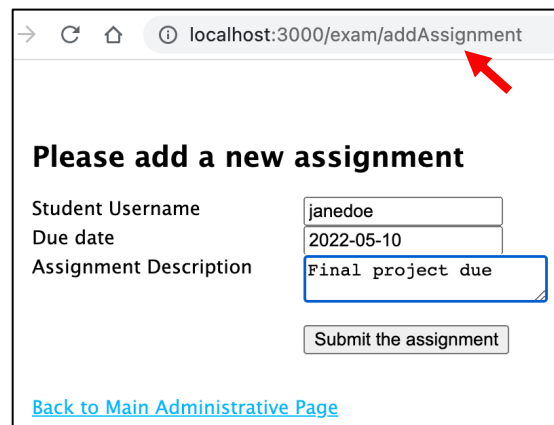


Figure 4

Grading Criteria

(8 points) Successfully downloaded starter code, uploaded to Heroku website and works

- Initial Project Setup
 - Create a new project: `express --no-view ExamII`
 - You will need to install proper packages. For example,

```
npm install pg dotenv bcryptjs passport passport-local express-flash cookie-session --save
```
 - Don't forget to define DATABASE_URL into .env to access the database.

Web Application Development: Exam II

- Starter Code

- Download examII.zip from BlackBoard and move these files to the appropriate directories. There are 9 files.
 - ExamII
 - app.js
 - ExamII/routes/
 - exam.js,
 - ExamII/public/
 - exam.html, admin.html, notAdmin.html addAssignment.html
 - ExamII/public/javascripts
 - message.js, notAdmin.js, whoami.js

- Database Setup

- You will need to create tables and insert rows (see below)

-- 2 tables: examusers, assignment

```
CREATE TABLE examusers ("id" serial primary key, username text, password text, isAdmin text);
```

```
CREATE TABLE assignment ("id" serial primary key, username text, description text, due date);
```

-- Password is hello. You may change it using bcrypt.js

```
INSERT INTO examusers (username,password,isAdmin)
```

```
VALUES('janedoe','$2a$10$tXMKF036p0ZYIxF/cJEHauw/TFrcho4DXy41Kt12D3Lbnzr221hmK','student');
```

```
INSERT INTO examusers (username,password,isAdmin)
```

```
VALUES('yoda','$2a$10$dxRF9y4FyqQ90M93xha6G01ZifqHUC/L3HI6KWoTJv91nI0c4lxfC','admin');
```

```
INSERT INTO assignment (username,description,due) VALUES('janedoe','Finish authentication exercise','2022-04-12');
```

```

jung=# select * from examusers;
 id | username | password | isAdmin
-----+-----+-----+-----
  1 | janedoe | $2a$10$tXMKF036p0ZYIxF/cJEHauw/TFrcho4DXy41Kt12D3Lbnzr221hmK | student
  2 | yoda    | $2a$10$dxRF9y4FyqQ90M93xha6G01ZifqHUC/L3HI6KWoTJv91nI0c4lxfC | admin
(2 rows)

jung=# select * from assignment;
 id | username | description | due
-----+-----+-----+-----
  1 | janedoe | Finish authentication exercise | 2022-04-12
(1 row)
```

- Test localhost, create an application on Heroku and Test it!

- Follow instructions how to publish to Heroku (Full instructions from <https://seikyung.com/>). For example,
 - git init
 - heroku login
 - heroku create
 - git remote -v
 - echo "node_modules" >> .gitignore
 - echo .env >> .gitignore
 - heroku addons:create heroku-postgresql:hobby-dev
 - more instructions...

- Success? You earned 8 points.

Web Application Development: Exam II

(12 points) you need to implement 3 features to receive the remaining points.

- **Feature1 (2 points): error checking (Figure 5)**
 - When creating a new assignment,
 - if the username doesn't exist, show a proper message – “User does not exist”
 - if successful, show a proper message – “Created assignment successfully”

Figure 5 shows three states of the 'Please add a new assignment' form:

- Initial State:** The form has fields for 'Student Username', 'Due date', and 'Assignment Description', and a 'Submit the assignment' button. A link 'Back to Main Administrative Page' is at the bottom.
- Success State:** After successful submission, a message 'Created assignment successfully' is displayed, along with the 'Back to Main Administrative Page' link.
- Error State:** If the username 'noUsers' is entered, an error message 'User does not exist' is shown, with the 'Back to Main Administrative Page' link.

Figure 5

Hint: Refer to code for how other error or success messages are handled.

- **Feature2 (5 points): Change Password (Figure 2 → Figure 6)**
 - You will need to implement a page to change passwords.
 - When changing password, if user's current password was wrong, show proper message.
 - If user's new password and retyped new password are different, show proper message.
 - Your implementation must be done in **server-side**

Hint: You will need to use “UPDATE” SQL to change to a new password:
`UPDATE examusers set password = $1 where username=$2`

Figure 6 shows three states of the 'Please type your new password' form:

- Initial State:** The form has fields for 'Current password', 'New Password', and 'New Password one more time', and a 'Change Password' button. A link 'Log out' is at the bottom.
- Success State:** After successful password change, a message 'You successfully changed password' is displayed, along with the 'Log out' link.
- Error State:** If the current password is wrong, an error message 'The two passwords you entered are not the same' is shown, with the 'Log out' link.

Figure 6

Web Application Development: Exam II

- **Feature3 (5 points): (Figure 3 → Figure 7)**
 - You will need to implement a page to create a new user (either admin or student)

The figure consists of three overlapping screenshots of a web application interface:

- Left Screenshot:** Shows the "Main Administrative Page" with a form for adding a user. The form includes fields for "user name" (containing "skywalker"), "Password" (containing "**"), and radio buttons for "student" (selected) and "admin". A "Submit the user information" button is at the bottom. A red arrow points to the browser address bar showing "localhost:3000/exam/addUser".
- Middle Screenshot:** Shows a success message "We created your account successfully!" above the same "Main Administrative Page" form.
- Right Screenshot:** Shows an error message "User exists" above the "Main Administrative Page" form. The "user name" field now contains "janedoe".

Each screenshot includes a "Back to Main Administrative Page" link at the bottom.

Figure 7

Hint: Refer to code for how I implemented "Add an assignment" page.

Web Application Development: Exam II

Due 6pm Tuesday, 4/19

- **Academic dishonesty**

- You may consult any books, online documentation, all class materials.
- You must NOT discuss the exam with anyone.
- You must NOT post questions online. For example, any website “helping assignment by asking questions”
- If you see exam questions online, you must notify instructor because that means somebody in class posted/asked questions online.
- You must NOT share solutions to anyone except instructor.
- You must REFUSE to help anyone who asks for your help.

- **How and what to submit?**

- Email your **Heroku URL** (<https://xxx.herokuapp.com/>) to me sjung@bridgew.edu
- Your email subject should be **COMP335: Exam submission**
- **readme.txt**
 - You must have at least “yoda” (password *hello*) and “janedoe” (password *hello*)
 - If you successfully implemented feature 3, provide at least two additional accounts (and passwords): one for **admin** and one for **student**.
 - Write a brief **description** regarding what’s working, and what’s not working.
- Attach the compressed file (only one file - **lastname.zip**). I can NOT accept any other file format.
 - ONLY compress the following source code:
readme.txt (testing credentials and description)
 - ExamII/routes/
 - exam.js,
 - ExamII/public/
 - exam.html, admin.html, notAdmin.html addAssignment.html
 - changePassword.html (for feature 2 – you may have different name)
 - addUser.html (for feature 3 – you may have different name)
 - ExamII/public/javascripts
 - message.js, notAdmin.js, whoami.js

- **Checklist!**

- Did you check your Heroku URL (<https://xxx.herokuapp.com>) work?
- Did you include readme.txt?
- Did you make **ZIP** only for xx files?
- Did you compress entire project? **NEVER DO IT!**
- Using non-zip format, such as .rar or .7z? **NEVER DO IT!**
- Is your file name lastname.ZIP?
- Email to Dr. Jung (sjung@bridgew.edu) no later than **6pm Tuesday 4/19**.

- **Grading – total 20 points**

- (8 points) Successfully downloaded starter code, uploaded to Heroku website and works.
- (2 points) Feature1 Error Checking
- (5 points) Feature2 Change Password
- (5 points) Feature3 Add a user (either admin or student)
- Any of these cases that I must communicate with student, the student will have -5 points
 - For example, file format is wrong, missing readme.txt, etc.
- No points (0 points), if it only works on your computer!
- No points (0 points), if your URL must include file extension. For example, exam.html

- **Help – Office hours**

- I’ll be available
 - 9:30 – 2pm Thursday 4/14 (First day of exam)
 - 9:30 – 2pm Tuesday 4/19 (Last day of exam)
- Feel free to email me. However, do not expect a quick response.
- Check BSU email for announcement.