

During the exam, you may reference printouts of any source code you have written (HTML, CSS, Javascript, SQL, PHP, Laravel). Write your name on this exam sheet and write your answers on the answer sheets provided.

1. (3 pts) Name a web framework and describe why you would use such a framework.
2. (2 pts) What is the path to get to the standard login controller in a Laravel application.
3. (3 pts) What does AJAX stand for and why would we use it in a web application?
4. (2 pts) Why might one use a REST API instead of directly querying the database?
5. (2 pts) Name two view engines from any framework we have studied this semester.
6. (2 pts) Name two response codes and their corresponding number.
7. (3 pts) What is the purpose of middleware and how would you find it in the Laravel framework?
8. (4 pts) What is the difference between client-side Javascript and NodeJS?
9. (2 pts) What is NPM used for in the context of NodeJS?
10. Suppose you are working on a web application built in the Laravel framework that allows users to enter data about movies. Further suppose that the page that allows entry of a new movie currently looks like this:

```
<form method="post" action="{{ url('/add_movie') }}">
  Title: <input type="text" name="title">
  <br>

  Genre: <select name="genre">

    <option value="1">Action</option>

    <option value="2">Comedy</option>

    <option value="3">Horror</option>

  </select>
  <br>

  Year: <input type="text" name="year">
  <br>
  <input type="submit" value="Save">
  <br>
</form>
```

- (a) (3 pts) What command would you add to the routes file in order to pass this form request to a function named `addMovie` controller called `MovieController`?
- (b) (6 pts) Write the `addMovie` function in the controller so that it validates the form. If any of the validations fail it should return to the original form. Otherwise it should write "Success!" to the error log. Validations to perform: title must exist and have no more than 200 characters; genre must exist and be 1, 2, or 3; year must exist and contain exactly 4 digits.
- (c) (6 pts) Show how you would modify the HTML code given above so that if validation fails, it will repopulate the form data for the user. Write your answer on the exam sheet to show your modifications rather than re-writing the entire form.
- (d) (2 pts) What changes would you need to make in order to provide CSRF protection for the "add movie" action described here.
- (e) (6 pts) Suppose you are creating a page that will display all movie data. Further suppose that the movie data has been passed to the view from the controller in a variable named `movies`. Further suppose that this variable is an array of objects with each object having this form:

```
{
  title: "Men in Black",
  genre: 2,
  year: 2010
}
```

Assuming the view is handled by a blade template file, show how you would display all movie data to the page while preventing XSS attacks. To format the data, just put the title, genre (no need to translate the number to a string), and year together on a line separated by a space. Each line should have only one movie.

11. For this question you will be doing similar steps as in the previous problem except we will assume you are working in the NodeJS framework. There are some tweaks, however. So ...

Suppose you are working on a web application built using the NodeJS framework that allows users to enter data about movies. Further suppose that the page that allows entry of a new movie currently looks like this:

```
<form method="post" action="/add_movie">
  Title: <input type="text" name="title">
  <br>

  Genre: <select name="genre">

    <option value="1">Action</option>
```

```

        <option value="2">Comedy</option>

        <option value="3">Horror</option>

    </select>
    <br>

    Year: <input type="text" name="year">
    <br>
    <input type="submit" value="Save">
    <br>
</form>

```

- (a) (6 pts) Write an `addMovie` controller function in the controller that will reside in a file named: `controllers/movie.js`. In the controller add code to validate the form. If any of the validations fail it should return to the original form and pass to the form data (via the session) to it. Otherwise it should write "Success!" to the console log. Validations to perform: title must exist and have no more than 200 characters; genre must exist and be 1, 2, or 3; year must exist and contain exactly 4 digits.
- (b) (4 pts) Assuming the `addMovie` function you created above is the first function to be written in `controllers/movie.js`, what commands would you add to the routes file in order to pass the form request to that function?
- (c) (6 pts) Show how you would modify the HTML code given above so that if validation fails, it will repopulate the form data for the user. Write your answer on the exam sheet to show your modifications rather than re-writing the entire form. You should assume the view is using the EJS templating format.
- (d) (6 pts) Suppose you are creating a page that will display all movie data. Further suppose that the movie data has been passed to the view from the controller in a variable named `movies`. Further suppose that this variable is an array of objects with each object having this form:

```

{
  title: "Men in Black",
  genre: 2,
  year: 2010
}

```

Assuming the view is handled by an EJS template file, show how you would display all movie data to the page while preventing XSS attacks. Use the same formatting as in the previous problem.