Full Stack Web Developer Nanodegree

Project Descriptions and Rubrics

Movie Trailer Website

Description

Supporting Course: Programming Foundations with Python
Students will write server-side code to store a list of their favorite movies, including box art imagery and a movie trailer URL. Students will then serve this data as a web page allowing visitors to review their movies and watch the trailers.

Rubric

Criteria	Does Not Meet Specifications	Meets Specifications	Exceeds Specifications (Completely Udacious)
Functionality	Page does not present required content (movie title, box art and trailer link).	Page does present required content (movie title, box art and trailer link).	Student includes additional information about their favorite movies (actors, release date, etc) or personalizes their page with additional HTML or CSS features.
	Page is not dynamically generated from a Python data structure.	Page is dynamically generated from a Python data structure.	*Not available.*
	Page contains errors.	Page is error free.	*Not available.*
Code Quality	Code formatting is not ready for personal review.	Code is ready for personal review and neatly formatted.	*Not available.*
Comments	Comments are not present in code.	Comments are present.	*Not available.*
	Existing comments do not effectively explain code segments.	Comments effectively explain longer code procedures.	*Not available.*
Documentation	No README file is included README file is incomplete.	A README file is included. README file includes details of all the steps required to successfully run the application.	*Not available.*

Tournament Results

Description

Supporting Course: Relational Databases

Students will develop a database and an application that will store player and game data. When given an input of player match-ups and win/loss information for a round of games, the application will properly process this data and provide the following round of match-ups and current standings of each player.

Rubric

Criteria	Does Not Meet Specifications	Meets Specifications	Exceeds Specifications (Completely Udacious)
Functionality	The module does not pass the included unit tests.	The module passes the included unit tests.	The module passes the unit tests; and separately, one or more of the extra credit options are implemented.
Table Design	Tables do not have meaningful names or there are unnecessary tables defined.	Tables have meaningful names and there are no unnecessary tables defined.	Tables have meaningful names and there are no unnecessary tables defined. In addition, views are used to make queries more concise.
Column Design	Table columns do not have meaningful names, are of the improper data type, or there are unnecessary columns defined.	Table columns have meaningful names, are of the proper data type, and there are no unnecessary columns defined.	Table columns have meaningful names, are of the proper data type and there are no unnecessary columns defined. In addition, primary and foreign keys are correctly specified.
Code Quality	Code formatting is not ready for personal review or features opportunity for SQL injection attacks.	Code is ready for personal review, is neatly formatted, and makes use of query parameters appropriately to protect against SQL injection attacks.	Code is ready for personal review, is neatly formatted and makes use of query parameters appropriately. All sorting and aggregation are performed in the database.
Comments	Comments are not present in code or existing comments do not effectively explain code segments.	Comments are present and effectively explain longer code procedures.	*Not available.*
Documentation	No README file is included or the file is incomplete.	A README file is included detailing all steps required to successfully run the application.	*Not available.*

Catalog App

Description

Supporting Courses: Full Stack Foundations, Authentication/Authorization, Relational Databases
Students will develop an application that provides a list of items within a variety of categories as well as provide a user registration and authentication system. Registered users will have the ability to post images and descriptions of items.

Rubric

Criteria	Does Not Meet Specifications	Meets Specifications	Exceeds Specifications (Completely Udacious)
API Endpoints	Page does not correctly implement a JSON endpoint with all required content.	Page does implement an JSON endpoint with all required content.	Student researches and implements additional API endpoints, such as RSS, Atom or XML.
CRUD: Read	Page does not read category and item information from a database.	Page does read category and item information from a database.	Student adds an item image field that is read from the database and displayed on the page.
CRUD: Create	Page does not include a form allowing users to add new items to the database table or does not correctly process submitted forms.	Page does include a form allowing users to add new items and correctly processes submitted forms.	Student updates the new item form to correctly process the inclusion of item images.
CRUD: Update	Page does not include a function to edit/update a current record in the database table or does not correctly process submitted forms.	Page does include a function to edit/update a current record in the database table and correctly processes submitted forms.	Student updates the edit/update item form to correctly process the inclusion of item images.
CRUD: Delete	Page does not include a function to delete a current record in the database table.	Page does include a function to delete a current record.	Student researches and implements this function using POST requests and nonces to prevent cross-site request forgeries (CSRF).
Authentication & Authorization	Page does not implement a third-party authentication & authorization service (like Google Accounts or Mozilla Persona); page implements its own, insecure authentication & authorization spec; or create, update, delete operations do not consider authorization status prior to execution.	Page does implement a third-party authentication & authorization service; and create, delete and update operations do consider authorization status prior to execution.	*Not available.*
Code Quality	Code formatting is not ready for personal review.	Code is ready for personal review and neatly formatted.	*Not available.*
Comments	Comments are not present in code or existing comments do not effectively explain code segments.	Comments are present and effectively explain longer code procedures.	*Not available.*
Documentation	No README file is included or the file is incomplete.	A README file is included detailing all steps required to successfully run the application.	*Not available.*

Conference Organization

Description

Supporting Courses: Developing Scalable Apps with Python
Students will develop an API server to support a provided conference organization application that exists on the web as well as a native Android application. The API supports the following functionality found within the app: user authentication, user profiles, conference information and various manners in which to query the data.

Rubric

Criteria	Does Not Meet Specifications	Meets Specifications	Exceeds Specifications (Completely Udacious)
App Architecture	App is not architected as a Web Service API.	App is architected as a Web Service API.	*Not available.*
	App can not support a variety of possible front-end clients.	App supports a variety of possible front-end clients.	*Not available.*
Task 1: Design Choices (Implementation)	Student does not add classes for Session and SessionForm.	Student adds classes for Session and SessionForm.	Student implements additional functionality such as an entity for speakers.
Task 1: Design Choices (Response)	The README file does not include an explanation about how sessions and speakers are implemented.	The README file includes an explanation about how sessions and speakers are implemented.	The README file includes an explanation of the design decisions behind the additional functionality.
	Student response does not justify the implementation details.	Student response shows understanding of the process of data modeling and justifies their implementation decisions for the chosen data types.	*Not available.*
Task 2: Session Wishlist	Users are not able to mark sessions they are interested in and/or retrieve their own current wishlist.	Users are able to mark sessions they are interested in and retrieve their own current wishlist.	*Not available.*
Task 3: Additional Queries	The README file does not describe two additional query types.	The README file describes two additional query types that are consistent with the goals of the project.	*Not available.*
	Additional queries are not implemented.	Both of the proposed queries are implemented.	*Not available.*
Task 3: Query Problem	Student does not explain the problem with the provided query in the README file.	In the README, student describes the reason for the problem with the provided query.	*Not available.*
	Student does not propose a solution to the problem in the README file.	In the README, student proposes one or more solutions to the problematic query.	Student has implemented the proposed solution(s) in actual code.
Task 4: Featured Speaker	Student does not implement getFeaturedSpeaker().	Student implements getFeaturedSpeaker().	*Not available.*
	Student does not use App Engine's Task Queue when implementing the featured speaker logic.	Student uses App Engine's Task Queue when implementing the featured speaker logic.	*Not available.*
Code Quality	Code formatting is not ready for personal review.	Code is ready for personal review and neatly formatted.	*Not available.*
Code Readability	Code does not follow an intuitive, easy-to-follow logical structure.	Code follows an intuitive, easy-to-follow logical structure.	*Not available.*

	Comments are not present in code or existing comments do not effectively explain code segments.	Comments are present and effectively explain longer code procedures.	*Not available.*
Documentation	No README file is included.	README file is included.	*Not available.*
	README file is incomplete.	The README file provides details of all the steps required to successfully run the application.	*Not available.*

Linux-based Server Configuration

Description

Support Course: Linux Administration Basics
Students will be provided a baseline Ubuntu virtual machine and must prepare that virtual machine to host a web application that has been provided. The virtual machine must be configured to protect against a variety of common attacks and host the web application so that it functions appropriately.

Rubric

Criteria	Does Not Meet Specifications	Meets Specifications	Exceeds Specifications (Completely Udacious)
User Management	Remote login of the root user is still enabled, a remote user that can sudo to root has not been defined, user passwords have not been changed.	Remote login of the root user has been disabled, a remote user that can sudo to root has been defined, user passwords are set securely.	*Not available.*
Security	Key-based SSH authentication has not been configured, SSH is still hosted on the default port, applications have not been updated to their most recent releases.	Key-based SSH authentication is enforced, SSH is hosted on a non-default port, applications have been updated to most recent updates.	The firewall has been configured to monitor for repeat unsuccessful login attempts and appropriately bans attackers; cron scripts have been included to automatically manage package updates.
Application Functionality	Web-server has not been configured to serve the provided application, database server has not been configured to serve data, VM can not be remotely logged into	Web-server has been configured to serve the provided application, database server has been configured to serve data, VM can be remotely logged into.	The VM includes monitoring applications that provide automated feedback on application availability status and/or system security alerts.
Configuration File Comments	Comments are not present in code or existing comments do not effectively explain code segments.	Comments are present and effectively explain longer code procedures.	*Not available.*
Documentation	No README file is included or the file is incomplete.	A README file is included detailing all steps required to successfully run the application.	*Not available.*