University of Nebraska Omaha

ISQA 3900 Web Application Development - Project

Design, Develop, and Deploy a Working Web Application as Defined Below

Scroll to page 2 or click here to review the - Required Deliverables

Objectives

- Team demonstrates the ability to be actively engaged in a development project.
- Team will create user stories for the features desired for the application.
- Team will begin to organize a backlog of "Must Have Features" and "Nice to Have Features" sorted by priority based on user stories.
- Team demonstrates effective development and management of a project backlog based on User Stories.
- Team demonstrates the ability to create and manage a release plan for a working website.
- Team members work with the product owner to prioritize the project backlog.
- Team delivers effective regular product owner demonstrations and project retrospectives.
- Team delivers and executes a comprehensive test plan for each sprint.
- Team creates an effective data model that meets the requirements of the project.
- Team effectively manages the code being developed by all team members using Git and Github code version control facilities.
- Team deploys a working application. All product demonstrations of new features occur from the remote server instance NOT from a group member's personal project.

Approach

Teams of up to 5 students will select a project to develop and deliver. The instructor will play the role of the product owner, but student groups are responsible for selecting a project application theme.

Requirements:

- Teams will be required to meet virtually or in person **AT LEAST TWICE PER WEEK** to conduct 'Stand-Up' meetings with at least 2 days between meetings.
    - Notes from EACH meeting must be recorded and saved to a shared folder for each meeting.
- At the end of each Sprint period, each team will present their work to the instructor.
- For Sprints after Sprint 0, a demonstration of the ***deployed*** version of the application must be included in the presentation.
- Each team member is expected to contribute to all parts of the assignment, including documentation and code development.
    - **Each team member is required to push updated python and/or HTML files to GitHub during each phase of the project.**
    - Do NOT rely on 1 or 2 group members to complete the technical work for this application.

Each Project Must Include the Following Basic Components:

- 4 or more programmer defined Database tables in addition to the database tables provided by Django.
- CRUD features for at least one database table through the application itself *in addition to* the ability to add database records through the admin panel.
- At least 2 types of users
- Several web pages (at least 4 linked pages)
- The ability to send e-mail notifications (This may be implemented as a password reset feature)
- Deployment to a server - PythonAnywhere or Heroku
- A shared folder for all Sprint Documents with folders for each Sprint shared with svlasnik@unomaha.edu
- A shared Trello Board to track work - Share with svlasnik@unomaha.edu
- A group GitHub repository – Shared with **svlasnik-uno**

Required Deliverables –

See also the project assessment rubric file posted to Canvas with this assignment for a detailed breakdown of the points possible and Presentation requirements.

Canvas Assignment Submission Requirements:

ONE MEMBER FROM EACH GROUP - SUBMIT THE FOLLOWING LINKS TO THIS ASSIGNMENT ON CANVAS FOR THIS SPRINT:

OneDrive folder link for your group (or other shared folder link)
Trello link for your project
GitHub link for your project
PythonAnywhere Deployed Application Link (Sprints 1, 2, and 3 only)
Ensure your presentation Power Point file is in the shared folder.
Deliverables:

ALL CODE WILL BE UPLOADED AND MANAGED WITH GITHUB for sprints 1-3.
ALL OTHER FILES WILL BE STORED IN OneDrive - or other shared folder - be sure to share with svlasnik@unomaha.edu
PROJECT MANAGEMENT INFORMATION WILL BE STORED IN TRELLO

Use a shared folder for documents, Trello for managing tasks, and GitHub for all application code

1. **Application and Organization Overview:** a description of the application and team organization
2. **Use Case Diagrams and User Stories**
3. **Data model** for the application. This can be created using the Oracle SQL Data Modeler or other modeling software. Updated version – each sprint.
4. **Detailed application mock-ups** of all screens to be delivered for the current sprint.
5. **Project release plan and backlog:** For Sprint 0, the release plan should be complete for the entire project and include all features planned for the project and when each feature will be delivered, grouped by Sprint number.
   a. Note: The release plan and backlog will be updated as needed as the project moves through subsequent sprints

6. **Burndown Chart:** Estimated hours/tasks and completed hours/tasks
7. **Potential Challenges and How to Overcome These Challenges**
8. **Notes from all 'stand-up' meetings – at least 2 held each week.**
9. **All Sprints after Sprint 0:**
    a. **Project test plan and results** of all tests run as well as bugs identified in the testing of this project. Defects should be tracked in GitHub.
    b. **Functioning web application that meets requirements of the product owner.**
        i. Each sprint after Sprint 0 must include new features implemented since the previous sprint. Conduct a live or recorded demo of the application during the presentation.
    c. Updated application code pushed to GitHub: GitHub repository updated with current version of the functioning application code.
    d. **Deployed application –** provide a link to the deployed application with a **superuser** account created with the following credentials:
        i. **username:** instructor
        ii. **password:** mavericks
    e. **Demo of deployed features during presentation.**
10. **Final Sprint: Documentation and Presentation:**
    a. Team Lessons Learned in the Project
    b. **Assessment of plan versus actual plan**, (see Project Assessment Rubric for final sprint for details)
    c. **Recorded Video** in shared folder of a Demonstration of 3 Automated Test Cases running against the most up-to-date version of the application.
11. **AT THE END OF EACH SPRINT – (Recorded or in-person) Formal Presentation – Presentation file posted to Canvas**:
    a. **Provide an overview of the features to be delivered** for the current sprint.
    b. **Includes a discussion and presentation of all items listed above (1-10)**
    c. **Links to all shared components:** Trello board, shared documents folder, GitHub repo posted to Canvas
    d. See an example presentation file with this assignment on Canvas.

Individual Deliverables:

1. Individual students will provide a **peer review and reflection** via Canvas Quizzes of each team member for each sprint
2. Based on feedback provided by fellow team members, individual students will earn participation points for each Sprint.

Continued next page.

Rubric and Grading:

See '**3900 PROJECT ASSESSMENT RUBRIC.XLSX'** posted to Canvas with this assignment for a detailed breakdown of the points possible and Presentation requirements.

Component	% of Overall Course Grade
Sprint Demonstrations – S0, S1, S2, S3	30%
Peer Review, Reflection, and Individual Project Points	15%
Total	45%

Red = Team Grade

Individual Grade

REQUIRED Tools

One of the major objectives of this assignment is to prepare students for the Capstone classes and other development courses. Use the following tools generally selected to develop new web development projects. These include:

Tool/Function	Specific Tool
General Documentation	A shared folder – share with the instructor svlasnik@unomaha.edu
Agile Project Management Tool	Trello
Development Server	Django with SQLite and Python
Integrated Development Environment	PyCharm
Database Modeling Tool	Teams may select a tool of their choice for Sprint 0. Django model visualization should be used for development sprints. SQL Oracle Data Modeler may be used for Sprint 0 – see resources module on Canvas for link and instructions.
Test and development – Deployment platform	Pythonanywhere or Heroku
Code Versioning and Team Collaboration	GitHub
Mockup/Prototyping Tools	Teams may select a tool of their choice
Other tools	Teams may select these as needed