CATALYST TECHNICAL ASSESSMENT CRITERIA

1. The main areas of assessment:

Each of the candidate / student is generally assessed in these four main areas;

1.1. Version Control (Git & GitHub)

1.2. Back end development (MongoDB/SqlLite/PostgreSQL, Nodejs/Django)

1.3. Front end development (HTML, CSS,JAVASCRIPT)

1.4. Project walk through and Testing (System Functionality Testing)

2. The break down and Overall marks per section:

Each of the main areas listed above is broken down into sub areas as below;

2.1. Version Control (GIT & GITHUB) - (10 Marks)

2.1.1. Using GIT to locally manage project versions (8 Marks)

2.1.2. Using GITHUB to remotely manage project versions (2 Marks)

2.2. Back end development (40 Marks)

2.2.1. Database Development (MongoDB / SqLite / PostgreSQL) - (10 Marks)

2.2.2. Javascript Runtime Environment setup (Nodejs / Python) - (10 Marks)

2.2.3. madabase Management - Object-Relational-Model (Mongoose / Django-ORM) - (10 Marks)

2.2.4. Server SetUp, Configuration and Connection (Express / Django) - (10 Marks)

2.3. Front end development (HTML, CSS,JAVASCRIPT) - (40 Marks)

2.3.1. UI Design (PUG or VUE, BOOTSTRAP) - (20 Marks)

2.3.2. Form Handling & Validation (JAVASCRIPT) - (20 Marks)

2.4. Project walk through and Testing (System Functionality Testing) - (10 Marks)

2.4.1. Project walk through (5 Marks)

2.4.2. System Functionality Testing (5 Marks)

3. The Awarding of marks (What we look for):

What is required or expected of a student to win the marks as in the sections above.

3.1. Version Control (GIT & GITHUB) - (10 Marks)

- Forking a repository - (1 Mark)
- Cloning a remote repository - (1 Mark)
- Ignoring files and directories - (1 Mark)
- Branching [Creating and switching between branches] - (1 Mark)
- Staging changes [rendering changes ready for a commit] - (1 Mark)
- Commiting changes [creating a loggable version] - (2 Marks)
- Merging changes into a specified branch - (1 Mark)
- Pushing changes to a remote repository - (1 Mark)
- Making a pull request - (1 Mark)

3.2. Back end development - (40 Marks)

3.2.1. Database Development (MongoDB / Sqlite / PostgreSQL) - (10 Marks)

- Environment setup (Downloading and Installation) - (3 Marks)
- Creating a Database - (2 Marks)
- Creating a Collections or Entities - (2 Marks)
- Creating Documents or Records - (3 Marks)

3.2.2. Javascript Runtime Environment setup (NODEJS) - (10 Marks)

- Downloading and installation - (2 Marks)
- Starting a Nodejs / Django Project - (3 Marks)
- Project initial configurations - (2 Marks)
- Installing and saving dependencies / requirements - (3 Marks)

3.2.3. Database Management - Object-Relational-Model - (10 Marks)

- Installation and setup - (2 Marks)
- Creating schemas - (3 Marks)

- Creating models - (3 Marks)
- Creating Documents or Records - (2 Marks)

3.2.4. Server SetUp, Configuration and Connection - (10 Marks)

- Installation and setup - (1 Marks)
- Server connection - (1 Marks)
- Creating routes - (4 Marks)
- Form Data Handling (Data Parsing) - (4 Marks)

3.3. Front end development (HTML, CSS,JAVASCRIPT) - (40 Marks)

3.3.1. UI Design (PUG or VUE or Django, BOOTSTRAP) - (20 Marks)
Note: Applicant can win bonus (2 Marks) for using bootstrap

Either PUG [If candidate used pug templating engine to develop views]

- Preparing pug environment (Downloading and installation) - (2 Marks)
- Setting pug as the default Templating engine for the application Express server - (2 Marks)
- Creating Essential and well defined Pug templates to be served as view resources - (2 Marks)
- Proper management of static files like images (Application Assets) - (2 Marks)
- Having a submittable form and setting valid form action(s) - (2 Marks)
- Using Appropriate Request Methods in forms - (2 Marks)
- Actual Template designs. Appropriate implementation of the UI designs (Accuracy) - (8 Marks)

OR VUE [If candidate used Vue SPA Framework to develop views]

- Preparing vue environment (Downloading and installation) - (2 Marks)
- Initializing a vuejs application for the project - (2 Marks)
- Creating Essential and well defined vue components for the Application - (2 Marks)

- Creating and using Router, Routes and Router-view and Router-links - (2 Marks)
- Proper management of static files like images (Application Assets) - (2 Marks)
- Importing, Registration and Usage of Components in other components - (2 Marks)
- Actual Template designs. Appropriate implementation of the UI designs (Accuracy) - (8 Marks)

OR Django [If candidate used Django templating engine to develop views]

- Preparing Django Templating Language environment - (2 Marks)
- Setting DTL / Jinja as the default Templating engine for the application - (2 Marks)
- Creating Essential and well defined Django templates to be served as view resources - (2 Marks)
- Proper management of static files like images (Application Assets) - (2 Marks)
- Having a submittable form and setting valid form action(s) - (2 Marks)
- Using Appropriate Request Methods in forms - (2 Marks)
- Actual Template designs. Appropriate implementation of the UI designs (Accuracy) - (8 Marks)

3.3.2. Form Handling & Validation (JAVASCRIPT/PYTHON) - (20 Marks)

- Using appropriate events - (2 Marks)
- Using appropriate event Listeners that match with the selected events - (2 Marks)
- Well defined and invoked event handler(s) - (2 Marks)
- Preventing unnecessary default behaviours of Forms and form elements - (2 Marks)
- Validation of all required fields - (8 Marks)
- Precise, informative, and meaningful error messages used on validated fields - (2 Marks)
- Form can be successfully submitted if all fields are valid (successful submission) - (2 Marks)

3.4. Project walk through and Testing (System Functionality Testing) - (10 Marks)

3.4.1. Project walk through (5 Marks)

- Concise and precise - (1 Mark)
- Articulative - (1 Mark)
- Accurate [right on point] - (1 Mark)
- Confident - (1 Mark)
- Convincing - (1 Mark)

3.4.2. System Functionality Testing (5 Marks)

- System Accuracy [Does it give the desired output]
- System Consistent [Can it accurately perform the same task and give the desired output over and over again]
- Fast [in terms of Performance]
- Error Free [No runtime errors]
- Intuitive [User needs less or No guidance to use the system]