**Automatic Form Generation**

**Team members –**

1. Name - Sonal Chavan

E-mail – [schavan6@student.gsu.edu](mailto:schavan6@student.gsu.edu)

1. Name – Charishma Marada

E-mail – [vmarada1@student.gsu.edu](mailto:vmarada1@student.gsu.edu)

**How to run the project –**

1. Run python file WebForms.py with command line argument of a JSON form.
2. If JSON has errors in it, they will be printed on console. For valid JSON – HTML, CSS, Python, JavaScript and SQL files will be generated.
3. Run ‘.sql’ file to generate MySQL database.
4. Run generated python file.
5. Run HTML files to make an entry/display all entered data.

**Project Details-**

Only file in the project – ‘WebForms.py’ takes the JSON input, validates it and calls a series of functions to generate 2 HTML files (to make an entry and to display data), JavaScript fie that handles requests from these 2 HTML files, Python file to which backend requests are sent through jQuery and an SQL script which generates a database and table names of appropriate names.

**Extra features other than file generation –**

As an extra feature, we have validated the JSON file before generating it. Following checks are present –

1. Element names should be unique across the forms.
2. If a multi-valued element ( checkbox or multiselect) is present, at least one key tag has to be present on one of the input elements ( to provide a primary key – foreign key relation in generated SQL tables).

**Technologies used –** Python REST, HTML, CSS, JavaScript, jQuery, JSON.

1. **Contributions by team members –**
2. Charishma Marada

* Generation of SQL script
* Generation of Python API that fetches data from SQL tables and returns it.
* JSON validation extra feature
* Testing and bug fixes

1. Sonal Chavan

* HTML files generation
* JavaScript file generation
* Generation of Python API that inserts the form data to SQL tables
* Testing and bug fixes