**Software Engineering Assignment-4  
Class 4 Analysis:  
Adapting Process Models:**

* To create or design any software the thing that we needs is a “Road map” or “Generic software Process”. With the help of this Road map or generic software process will help us in successful software driving.
* In general scenario, every project is different and every team in different like all teams cannot work on the same module or the same component.
* No single software engineering framework is appropriate for every software product.
* Whatever the road map or generic process model we are following should be the best industry practices. So that the customer can get the best.
* Developers and stakeholders adapt generic process models and tailor them to fit the current project, skills of the team members and the user needs.

**Principles for Organizing Software Projects:**

* It will be risky to use a linear process model without ample feedback.
* It is never possible nor desirable to plan big up-front requirements gathering.
* Even if we have a big up-front requirement gathering it may not reduce the costs or prevent time slippage.
* Appropriate project management is integral to software development.
* Documents should be evolved with the software, it should not delay the start of construction.
* Stakeholders should be involved early and frequently in the development process.
* Tester also need to be involved in the process prior to software construction so that tester’s involvement can be seen in early stage.

**Characteristics of Agile Process Models:**

* Agile is not suitable for large high-risk or mission critical projects.
* Here we have minimal rules and minimal documentation
* Tester were involved in every phase like backlog refinement, sprint and every phases
* Product changes are easily accommodate.
* Depends heabily on stakeholders interaction like in sprint review, backlog refinement, sprint retrospective
* It is easy to manage
* Early delivery of partial solutions like easy user story can be selected and designed
* Informal risk management.
* Built in continuous process improvement.

**Characteristics of Spiral Process Models:**

* This process is not suitable for small, low-risk projects.
* Many steps are required along with the documentation.
* Tester are also involved in early stage.
* It is not easy to accommodate product changes until the prototype completed fully.
* Stakeholder involvement can be identified in planning and risk assessment.
* Require formal project management and coordination.
* Project end is not always abvious.
* Good risk management can be identified here.
* Process improvement handled at end of project.

**Agile Requirements Definition:**

* Stakeholder participation in active by matching their availability and valuing their input.
* Use simple models to reduce barriers to participation.
* Requirement representation techniques before using them will take time.
* Adopt stakeholder terminology and avoid technical understanding whenever possible.
* Breadth-first approach is used to get the big picture of the project.
* Developers and stakeholders refine requirements “just in time” as user stories are ready to be implemented
* User stories are prioritized so that they can be developed first.
* Questions are need to maintain models and documents not referred to in the future.
* Ensure support for stakeholder and resources availability during the requirements definitions.

**Resource Estimation for Agile spiral Model:**

* Team should use historic data to develop an estimate of number of days needed to complete each of user stories known at the start of the project.
* Organize the user stories into sets that will make up each sprint planned to complete a prototype.
* Total sprint duration is provide as an estimation for duration of the total project
* Revise the estimate as requirements are added to the project or prototypes are delivered.

**First prototype Guidelines:**

* Transition from paper prototype to software design
* Prototype a user interface
* Create a virtual prototype
* Add input and output to your prototype
* Test your prototype
* Prototype deployment

**Prototype Evaluation:**

* Test your prototype on the right people
* Ask the right questions
* Allow user to contribute ideas.

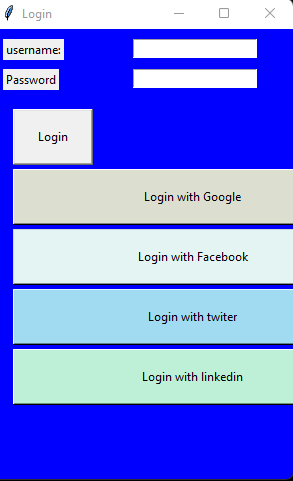
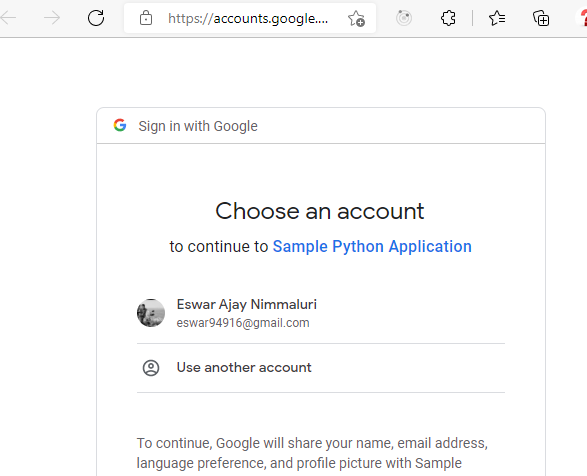
**Software release Maintenance:  
  
Maintenance:** Even after software is released and accepted the maintenance activities need to be done

Corrective Maintenance: reactive modification of software to repair problems discovered after the software has been delivered

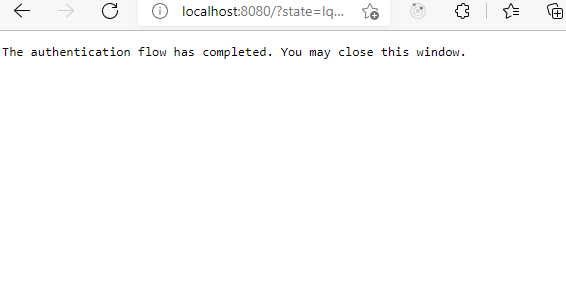
Adaptive Maintenance: reactive modification of software after delivery to keep software usable change in environment

Perfective maintenance: Better program code structure or improved documentation after software delivery

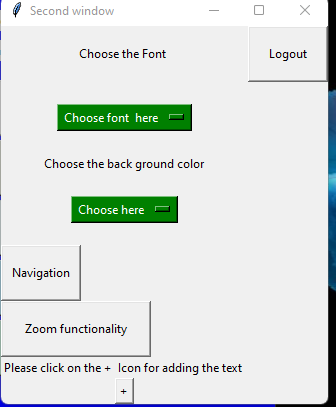
Preventive maintenance: proactive modification software after delivery to correct product faults before discovery by users.

**In Class Exercise:** **Click on login with Google:**

**Click on the gmail for login:**

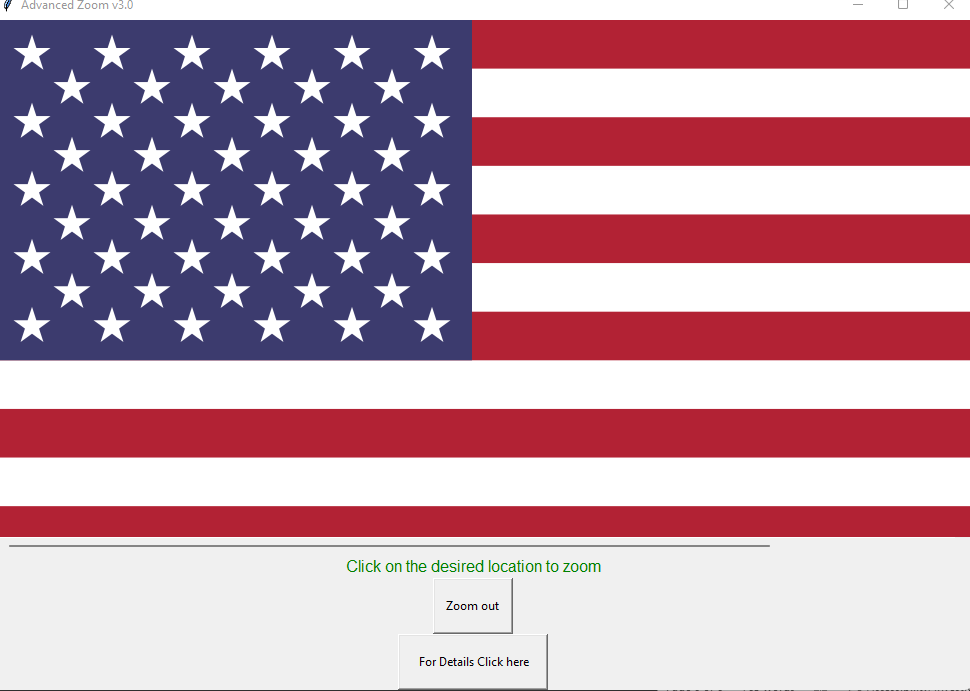
****

**After Successful login it open the internal Application:**

****

**With the previous functionality a new new feature is to be added to the application:**

The new new feature is we can able to see the us flag and ZOOM IN and ZOOM OUT functionality should work:



On click on the details link it should navigate to the link where we can get the details of the US flag:  
  


Here we can observe that a new feature is added to the existing prototype. Like previously we have developed an application where we are login using google authentication and after successful login user can be able to change the font, change the background , add the text box , navigate to new screens .  
  
Now user or the stakeholder requested that he required a new feature when he can be able to see the US Flag and perform zoom in and zoom out functionality. Also a new hyperlink or button should present, which should navigate to external url and customer can get the information about the US flag.