**Project Paper**

* Introduction…
  + Names of group members, the class name listed “CMPT475/477 – CS/IT/IS Capping Project”, and the name of the project “Credit Checker” or whatever you are calling it
    - Introduce the group and talk about each person’s role and how we split things up. Talk about HERBY and why we named it Foxify.
  + Executive summary of the project – single paragraph should be enough
* Analysis…
  + From the Exec summary, you can flow right into the user requirements
    - Complete documentation of functional and nonfunctional requirements as your team understood them along with any assumptions you made. This can be bulleted lists, but should have enough prose to make it readable
      * Talk about the user requirements that Kevin (IT) put together. Explain how he came to get those requirements.
  + Use case diagrams and any supporting documentation-
  + Activity diagrams and any supporting documentation- create a couple of activity diagrams.
* Plan…
  + Original project plan and your cumulative project status reports – you don’t necessarily need them all for the final paper, but what you want are the significant updates and salient points from each so that you have a record of how the project progressed – think of it like a diary of the project.
* Cost / ROI…
  + Cost report of development project from inception through prototype phase
  + Cost estimate to take prototype to production level
    - include testing, and implementation deployment
  + Cost estimate for 1 year of operation / maintenance
* Application Design…
  + Database design, ER model, and supporting documentation – tell someone how your database is designed .. so that if they had to, they could recreate it
    - Include sample SQL statements if you think that would help
    - Include instructions for importing initial data if applicable
  + User interface design,
    - Include all relevant screen shots and supporting documentation or explanations. This section should logically walk the reader through all views of your system, with visual support as well as text.
* Infrastructure Design…
  + IT requirements documentation that allows someone to build your environment again after you are gone – all VMWare instructions, ip addresses, userids, permissions, software levels, etc.
    - IT server configuration information (diagram if appropriate)
    - Network diagram if appropriate
* The project itself…
  + Link to Github or code repository where your code can be viewed by me
  + Link to active web portal demo running on IT Infrastructure in the Lab
  + Three test cases and instructions on how to run them against your application.
    - The cases must be different and should include extreme/boundary conditions, e.g., stupid users that type nonsense things where they shouldn't, regular users, and malicious users, etc.
* User documentation…
  + These can be printouts of your help pages OR they can be .doc files with the help information that would either be published as online help, in a printed guide, or later added as help pages inside the application
    - main “help” accessible from the application for end users (i.e. perspective students, parents, admissions counselors, etc.)
    - “help” accessible from the application for back-end users (i.e. Marist Admissions performing inquiry response and report processing)
    - administrator users guide for use in performing admin tasks (i.e. database updates, reports, course updates, affiliate approvals, etc.)
    - code maintenance guide – documentation that helps the “next team” understand how the code / database / web front-end components were intended to be updated, enhanced, ported, or whatever else can help take this project to production.
* Appendices as your team deems necessary

serious intro first- following ROI

then fun beginning- the good thing about - profiting from student developing effort

free labor getting something that you need

saving labor costs

more importantly making it easier from trans studs.