Team Name: APCS (Apple Peeler Corer Slicer)

Members:

NAMES	GITHUB USERNAMES
1) Matthew Skogen	https://github.com/mskogen
2) Ryan Davis	https://github.com/ryan-s-davis
3) Asher Farr	https://github.com/BillFlan
4) Chad Di Lauro	https://github.com/chaddilauro
5) John Henry Fitzgerald	https://github.com/jhff24
6) Kamiar Coffey	https://github.com/kamiarcoffey

Description:
APCS will be developing a web app that will give users data and recommendations about which ski resort is best to go to on any particular day. The decision algorithm will take into account the weather, day of the week, traffic, distance to the resort, size of the resort, usual crowds at the resort, etc. These factors will be compared to the user's personal preferences and ski pass to recommend the best resort for their desires. Each user will be able to save a profile containing all their preferences and potentially an email address so that they may receive a notification if a particular day matches their preferences especially well. This can also be expanded to destinations outside of Colorado on the Epic/Ikon passes to alert people when the best time to plan a large trip would be.

This product will allow users to maximize their time on the mountain and will create an overall better experience for all skiers. On a large scale, this will also help to spread out the times that people are heading out to the mountains for the winter and will reduce the total amount of traffic and travel time to the ski resorts in Colorado.

Vision Statement:
APCS will create a web app product that will make all its users' ski trip experiences better in ways that specifically tailor to each user's preferences.

Motivation:
APCS is a team of skiers and riders who are all passionate about building a tool that will make the difficult decision of which mountain to go to. APCS is excited to provide a much more informed decision than users would be able to make without this product.

The team is also excited to experiment with javascript and node.js, among other web development platforms. A particular source of motivation for the project is to gain experience using APIs to host outside information from databases on a web platform.

Risks:

The main risk that this team will face in putting together this application is the access to relevant data. The team has yet to finish investigating sources for weather, traffic, and snow data, which will be crucial to the functionality of this tool.

Additionally, the team is planning to learn a large number of new practices to complete this project. While APCS feels that this will be an attainable goal, it will be a large challenge and could put the overall project into jeopardy.

Risk Mitigation Plan:
One of the initial steps this team takes to build this project will be to investigate documentation and pricing plans for accessing this type of data from various sources. We will look to Google Maps, North American Aerospace Defense Command (NORAD), and National Weather Service first, and discuss alternative sources after further research and investigation.

The first approach will be to use APIs to run the app with current weather data. However, one possible alternative would be to use historical data to produce an estimate if the use of APIs becomes too difficult or time-consuming.

The initial focus will be to build a web-app that is compatible with mobile devices. A possible plan to descope if difficulties are encountered will be to just build a website. This should be highly attainable contingency given our choice of language and architecture.

Version Control:
- https://github.com/mskogen/APCS-meetingLogs
- https://github.com/mskogen/APCS-milestones
- https://github.com/mskogen/APCS-code-stuff

Verification of collaboration from all group members:

Development Method:
The team will make use of the agile/scrum methodology in order to complete this project as effectively as possible. The team will conduct a variety of sprints to meet project milestones.

Collaboration Tool:
Google collaboration suite & GroupMe

Proposed Architecture:
The preliminary structure for implementing this web-app is to use APIs from the National Weather Service, corroborated with individual ski resort websites to pull in data on recent and trailing snowfall.

Many resorts host data on their websites including live camera footage of base and summit conditions, on-the-ground reports of base snow depth, number of runs open (including by category), and recent snowfall (usually last trailing 24 hours, and last two weeks).

Likely Languages:
- HTML
- CSS
- Node.js with bootstrap

Database Resources:
- http://www.myweather2.com/developer/apis.aspx?uref=f8add9ec-fe95-4001-b822-df8680e49dce
- https://developers.google.com/maps/documentation/javascript/examples/

App Names:
- blizzardwizzard.com
- blizzardkicker.com