Teammates: Jesus Ayala, Neftali Garcia, Kaycee Garcia

- **Jesus Ayala**
- **Component type** (e.g., DLL, User Control, Cookie, etc.)
- **Description:** what it does, input/output
- **TryIt functionality** for each component, so the grader can **test** it
- **<u>Kaycee Garcia</u>**
- **Component type:** User Control, Session
- **Description:** My service lets the user choose from a drop down menu of all 50 states and keeps a log of all of the states chosen and outputs their average wind intensity.

Percentage of overall contribution: Member 1 :33% Member 2 name:33% Member 3 Neftali: 33%

Provider name	Page and component type, e.g., aspx, DLL, SVC, etc	Component description Does the component do are inputs /parameters and output/return value?	Actual resources and methods used to implement the component and where this component is used.
Neftali Garcia	Default.aspx ASPX page with server controls	TryIt page that allows testing Weather service and password hash DLL	Designed GUI with textboxes, buttons, and labels. Buttons in code-behind call the Weather5Day service and hashing DLL

Neftali Garcia	Global.asax	Global application start event logger	Captures and stores app startup time in Application["StartTime"], shown on GUI
Neftali Garcia	DLL (Class Library - SecurityTools.dll)	Local component that hashes and encrypts/decrypts a password string	HashPassword() + optional encryption added. Integrated into TryIt GUI via button
Neftali Garcia	WCF SVC (Service1.svc)	Web service: Weather5Day(string zipcode) Returns a 5-day forecast for a given ZIP code	Integrated OpenWeatherMap API to retrieve real weather data. Validates ZIP and returns forecast array. Deployed on WebStrar
Neftali Garcia	Service Reference	Service reference and WebClient logic to call WCF service	Added WeatherServiceRef to Web App. Called via Service1Client from Default.aspx.cs
Jesus Ayala	ASPX page & server controls	**Default.aspx**: Public landing page with app intro, Member/Staff buttons, and the Service Directory. No inputs; renders HTML UI and links.	Implemented in **ExpenseTrackerApp** project: Default.aspx markup and code-behind. Uses <asp:Button> controls for navigation and an HTML <table> for the directory.
Jesus Ayala	RESTful Service (Web API)	**SolarIntensity(lat, lon)**: Returns annual average solar insolation. Inputs: decimal latitude, decimal longitude. Output: decimal.	Implemented in **SolarService** project: Controllers/SolarController.cs using ASP .NET Web API attribute routing. Calls NASA POWER API via HttpClient, parses JSON with Newtonsoft.Json.

Jesus Ayala	DLL Function (SolarUtils.dll)	**EstimateAnnualOutput(dailyIntensity, panelSize)**: Calculates annual kWh output. Inputs: decimal dailyIntensity, decimal panelSize. Output: decimal kWh.	Implemented in **SolarUtils** class-library project (PanelCalculator.cs). Exposed static method and referenced by ExpenseTrackerApp. Invoked in TryIt_Solar.aspx.cs to compute final output.
Jesus Ayala	Cookie	**PreferredSize: Stores user's preferred panel size (kW). Input: panel size. Output: HTTP cookie persisted 30 days.**	Implemented in **ExpenseTrackerApp** TryIt_Solar.aspx.cs: creates HttpCookie("PreferredSize", size) on button click; reads it in Page_Load to prefill panel-size textbox.