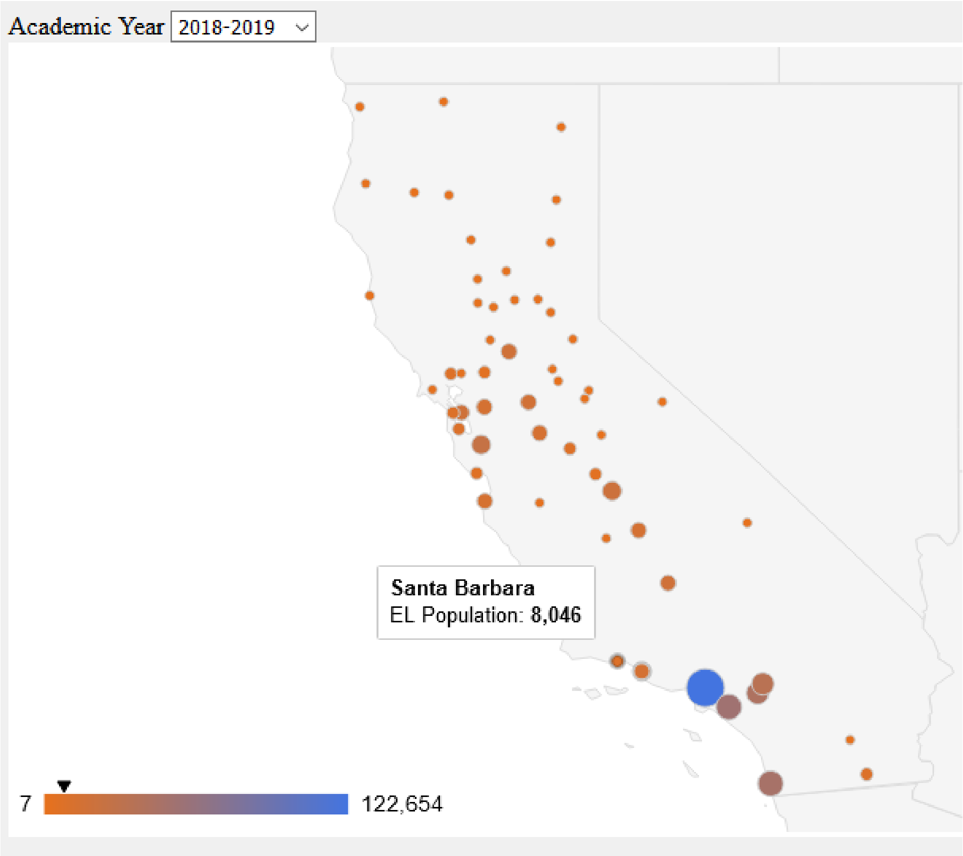
COMPUTER INFORMATION SYSTEM TO SUPPORT CALIFORNIA’S MIGRANT EDUCATION PROGRAM (MEP)



Academic Year: 2018-2019

AUTHOR: **MICHAEL CRAIG**

DATE: *September 14, 2019*

WESTERN GOVERNORS UNIVERSITY

CONTENTS

SECTION A - LETTER OF TRANSMITTAL

Michael Craig, Information System Analyst

1133 Deer Trail Lane, Solvang, CA 93463

9-15-2019

Subject: Decision-Support Application for Long-Term English Learner Population

To whom this may concern,

California public schools need an assistive function to process state-provided data on students in the Migrant Education Program.

The following proposal offers a solution that will transform millions of records into an informative dashboard. Data to be used is collected by an existing process owned by California's Longitudinal Pupil Achievement Data System (CALPADS).

The solution would support Business, Migrant Education, and Human Resource departments across California's K-12 schools; resulting value includes improved accuracy of staffing projections, mitigating the risk of over-staffing and saving taxpayer-funded budgets upwards of tens of millions of dollars.

The solution would be inexpensive to deploy compared to its potential value. School districts with existing equipment may spend close to $5000, whereas equipment costs may result in total expenses not exceeding $8000.

Please consider adopting this solution within your local district.

Thank you,

Michael Craig

Student ID: 000955248

Information System Analyst

1133 Deer Trail Lane, Solvang, CA, 93463

805-345-8100

PROJECT PROPOSAL

PROBLEM

The State of California is mandated by United States Federal law to provide the Migrant Education Program (MEP) as a service through a State Service Delivery Plan (SSDP)[1]. A Local Education Authority (LEA) may identify qualifying students, who are classified with the status of English Learner (EL); these students are reported and tracked through the California Longitudinal Pupil Achievement Data System (CALPADS) for the purpose of tracking their progression in the MEP.

In general, it is difficult to track movements of student bodies between schools; e.g. the matriculation of students from elementary schools into secondary education. Students participating in MEP require specialized teachers that may be difficult to find and hire. Human Resource departments across the state need accurate and reliable information as early as possible to support their decisions on fulfilling these staffing requirements.

VALUE

The proposed system would support Human Resource and MEP departments in California's K-12 Public Schools by providing historical trends of EL student data. Both departments may utilize the system to view the progression of EL student populations in feeder districts to anticipate and plan future staffing requirements for their local MEP.

Accurate staffing projections help avoid situations of under-staffing and hindrance of student progress. In contrast, over-staffing harms entire districts by incurring unnecessary expenses, reducing funds to improve or provide educational resources, and deterring talent from seeking future employment opportunities.

SYSTEM OUTLINE

The proposed solution is a website with an intuitive set of controls and charts that adapt to users' needs. The website may have restricted access where only internal users may view its information. Data for the system is already collected, cleaned, and provided by CALPADS and requires no additional effort by users.

DATASET

Since the state first adopted CALPADS, public school districts have annually reported the status of MEP students on the first Wednesday of October each year. In June, the state Department of Education distributes a file including every EL student in California as from the preceding October. The current set of records includes the last five years of data reported to CALPADS; an average of 500,000 new EL records have been produced each year.

HYPOTHESIS, GOAL, & OBJECTIVES

Hypothesis: The data utilized in the proposed solution may produce enough information to accurately support projecting staffing requirements for the Migrant Education Program.

Goal: To provide a system of functions capable of adapting to the business needs of California public school departments of Business, Human Resources, and Migrant Education.

Objectives:

- Develop a user-friendly dashboard that provides an intuitive display of the information.
- Allow flexible application design for fast and easy modifications, such as additional charts.
- Include self-adjusting capabilities to efficiently pull data that's been previously queried.

METHODOLOGY

The project's design will be heavily influenced by customer feedback and the expectancy of evolving acceptance requirements. Development teams will be in regular contact with customers and stakeholders to discuss progress and ensure that the project adheres and adapts to expectations. Documentation will be written by the principle of just good enough so that efforts can be focused on development.

FUNDING

Funding requirements may be broken down based on estimated resources required; initial development costs and long-term maintenance costs may be negotiated or non-existent.

Item	Units	Expense
Host Computer	1	$2,000
Development Software	1	$200
Database Software	1	$931
Research & Development hours	30	$150
Total		$7,631

STAKEHOLDER IMPACT

Adopting this solution will benefit stakeholders by improving the accuracy of planned expenses, providing immediate resources to students as needed, and streamlining processes associated with the matriculation of new students.

Unplanned expenses risk exceeding fiscal budgets, potentially impacting all current projects that are underway. Refining or establishing processes can improve resource efficiency and overall work output. Student stakeholders are limited to receiving benefits while they are attending school; ensuring resources are available as soon as they're needed will maximize value received by students during their attendance.

LEGAL

The Long-Term English Learner (LTEL) file provided by the state includes data that is recognized as Personally Identifiable Information (PII) in the Family Education Rights to Privacy Act (FERPA)[3]. The file also includes what is considered Directory Information, which may be not be disclosed without consent by a student over the age of 18, or consent from their parent or guardian.

Per the director of the United States Department of Education's Family Policy Compliance Office (FPC), under the circumstances in which educational records are anonymized in a way that the identity of a student cannot be determined (*Rooker, 2004*). The four-year dataset to be used in development has been scrubbed of any PII and conforms to the requirements outlined by the director in his 2004 letter.

Districts whom do not scrub PII from their datasets may avoid exposing any PII to the general public by deploying this solution in-house on a Local Area Network (LAN); Network security standards will need to adhere to industry best-practices. If this information would be preferred to be publicly available, measures must be taken to anonymize the dataset of any PII prior to loading the system.

EXPERIENCE

Michael Craig is an Information Systems Analyst for a California high school district; this solution is his capstone project for earning a Baccalaureate of Science in Computer Science from Western Governor's University. He has four years of experience supporting Software as a Service (SaaS) products for enterprise and VIP customers, along with director-level personnel in an international organization; he is currently coordinating state-mandated reporting efforts of teams spanned across multiple sites. He has experience developing standalone applications and scripts in Java, Python, C++, PHP, and Powershell; additional experience includes project management and developing software solutions that integrate with vendor-provided web applications and relational databases.

SECTION B – EXECUTIVE SUMMARY

GOAL: DECISION-SUPPORT

California public schools need an application capable of tracking data trends associated with English Learning students. A solution that can provide this information would support decisions concerning the business needs of their Business, Human Resource, and Migrant Education departments.

Objectives include:

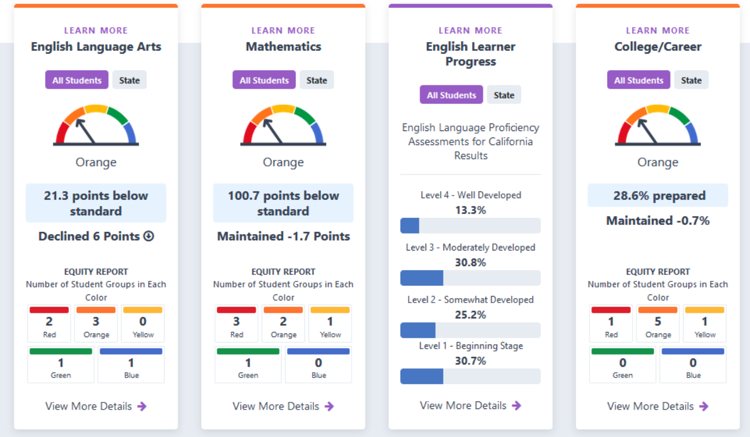
- An intuitive interface for non-technical users to navigate and read.
- Modular charts capable of adjusting by technical specifications and user choices.
- Supervised learning functionality that increases application efficiency over time.

CUSTOMER

This project targets director-level personnel of California's public schools: individuals who would use this information to support business decisions may impact the spending of public funds and education quality of students learning the English language. Users are expected to be non-technical subject matter experts in their respective fields.

PREDECESSOR

The California Longitudinal Aptitude Data System (CALPADS) offers aggregated statistics through reporting tools that are limited to annual views and do not offer year-by-year comparisons. The California School Dashboard, which is populated with data from CALPADS, visualizes data in the form of simple gauges that only span 5 degrees of a state[5]. Both tools are limited in allowing a broad audience of specialists to analyze patterns without significant, manual effort.



Screenshot from caschooldashboard.org (9/29/2019)

DATASET

The data included in this project is provided by CALPADS in the form of a comma-delimited csv file. The set includes a total of 2.5 million records collected by California public and charter schools over the span of four years. Students are tracked by unique identifiers, which have been masked to protect the privacy of students and avoid violating federal law; there is no map or key to reproduce the original state identifier.

Other fields included in each record contain birthdate of student, grade at the time of record collection, date of first enrollment in a

Field Name	Datatype
District	String
School	String
Student State ID	Int
Birthdate	Date
Years in MEP	Int
Grade	String
EL Status	String
First US K-12 enrollment	Date

US public school, years of participation in the English Learning program, status in the English Learning program, school of enrollment, school state identifier (CDS code), and district of enrollment.

To further protect student privacy, both dates have been translated to integer values representing the number of years since the date.

An approximate count of 500,000 new records are expected to be annually added in subsequent years.

DEVELOPMENT METHODOLOGY

An Agile methodology will be adopted for this project with an emphasis on object-oriented design. Project requirements are expected to change based on customer feedback as chart modules are produced and tested; communication with customers will be ongoing throughout each phase of development.

Database architecture will adhere to normalization processes that promote minimizing data redundancies and enforcing data integrity.

DELIVERABLES

- Database server configuration and architecture
- Webserver configuration
- Fully developed website using PHP

IMPLEMENTATION PLAN

1. Wrangle datasets from 2015 through 2018 for load preparation
    a. Hash state identifiers
    b. Convert date values to integers representing year counts
2. Design database architecture
3. Configure development environment
    a. Deploy new Ubuntu database server
        i. Install MS SQL Server 2016
            1. Create service account
            2. Create database architecture
            3. Load datasets
    b. Deploy new Ubuntu web server
        i. Install Apache2
        ii. Install PHP
        iii. Install SQL Server ODBC Driver
            1. Configure settings with database server specifications

      c. Configure PyCharm with SFTP synch to web server's web directory (/var/www/html/)

4. Develop Storyboard
   a. Consult with stakeholders to ensure requirements are met
   b. Develop mock-up of site
   c. Develop php modules for producing charts
      i. Load modules into site mock-up
      ii. Conduct iterative white and black-box testing
   d. Draft initial CSS template for customer review
      i. Update per customer specification
5. Deploy servers within customer environment

ACCEPTANCE TESTING

VALIDATION

Data projections produced by the application will be compared to local information systems. Discrepancies between counts may be explained by anomalies during the state reporting process, however confidence will be lost in the product if the data is determined to be invalid. Local record counts of EL students will be compared with the count provided by the state: validation will pass if less than 10% disparity exists between counts.

VERIFICATION

Unit testing will be conducted to ensure accuracy of aggregations; results must produce expectations with 100% consistency.

DEVELOPMENT ENVIRONMENT

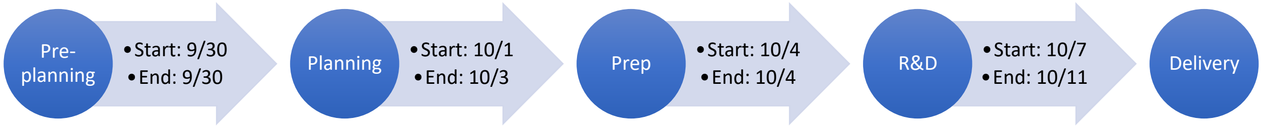
DESCRIPTION

Pycharm was chosen as the Integrated Development Environment (IDE) for this project. Pycharm *Professional Edition* includes the feature to establish a secure file transport (SFTP) connection to the remote web host and synchronize work directories during development. This configuration allows for stage-level testing as new code is produced.

EXPENSES

Item	**Units**	**Expense**
Hypervisor Host	1	$2,000
PyCharm Professional IDE	1	$200
Microsoft SQL Server 2016 Standard	1	$931
Programming and Testing hours	30	$150
Deployment hours	8	$25
Total		$7,831

TIMELINE



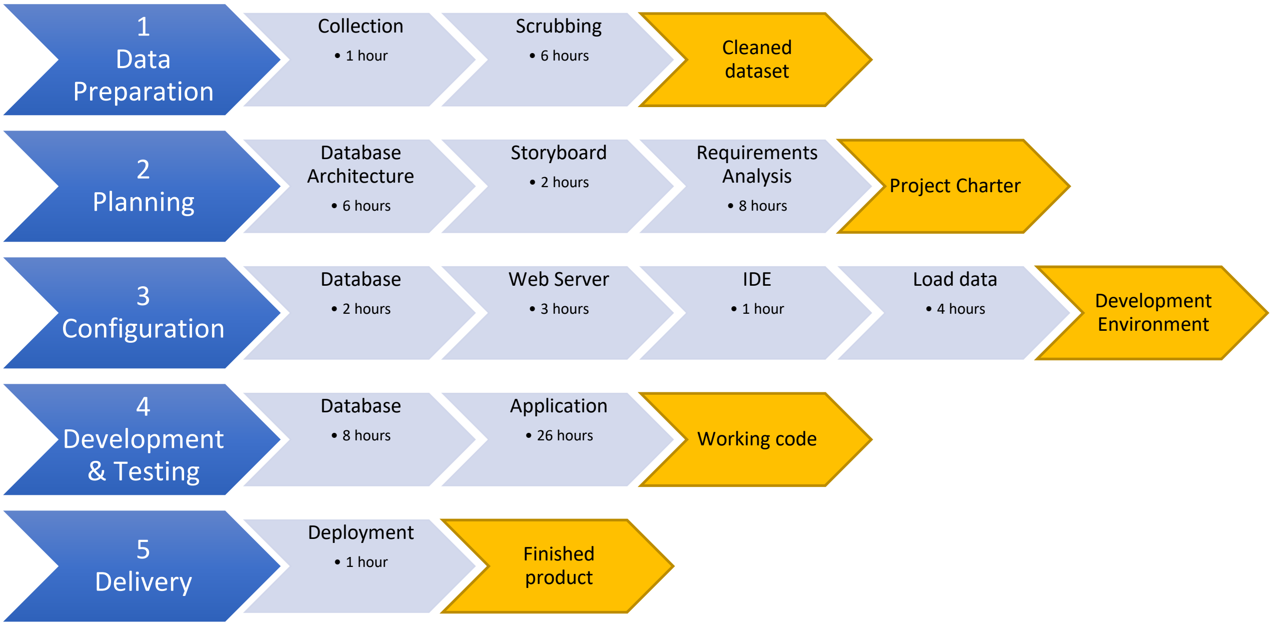
Pre-planning
- Start: 9/30
- End: 9/30

Planning
- Start: 10/1
- End: 10/3

Prep
- Start: 10/4
- End: 10/4

R&D
- Start: 10/7
- End: 10/11

Delivery



1 Data Preparation
- Collection
  - 1 hour
- Scrubbing
  - 6 hours
- Cleaned dataset

2 Planning
- Database Architecture
  - 6 hours
- Storyboard
  - 2 hours
- Requirements Analysis
  - 8 hours
- Project Charter

3 Configuration
- Database
  - 2 hours
- Web Server
  - 3 hours
- IDE
  - 1 hour
- Load data
  - 4 hours
- Development Environment

4 Development & Testing
- Database
  - 8 hours
- Application
  - 26 hours
- Working code

5 Delivery
- Deployment
  - 1 hour
- Finished product

SECTION C – CODE ANALYSIS

METHODS

Interactive queries are used in conjunction with the Google Charts API to create two analytical methods. Buttons are provided to users with a set of options queried from the database. An AJAX call posts the selected option to the application's controller, prompting the chart's data model to update. Variable selection is followed as a parameterized stored procedure using the selection in its WHERE clause.

DESCRIPTIVE

Variable selection is determined by user selection, which triggers the model to call a stored procedure that aggregates subset counts of students. The current product contains chart applications designed to aggregate totals by county of residence or school year.

PREDICTIVE

Linear, exponential, and polynomial regression methods have been applied to each visualization. The predictor function, H(x), is displayed in the browser’s console; H(x) may be used to calculate estimated predictions for future values.

DECISION SUPPORT

The descriptive method provides historical context that enables the predictive method. The predictor function can then be used to forecast EL student populations in upcoming school years.

DATASET

DESCRIPTION

The size of the raw csv flat file is 258 megabytes and contains over 250 million comma-delimited rows.

Field Name	**Datatype**
District	String
School	String
Student State ID	Int
Age	Int
Years in MEP	Int
Grade	String
EL Status	String
Years since first US K-12 enrollment	Int

PREPARATION

For the development of this solution, the dataset was cleaned in a way to remove any personally identifiable information (PII). State identifiers for students were replaced with a linear sequence with no key to map students to the original identifier. Dates, such as birthdays, were replaced with the number of years from a recent date.

Importing was done through SQL Server Management Studio’s (SSMS) built-in tool to import records from a flat file into a single table. Unfortunately, some school names had one to five commas within the name; this caused several fields to be inserted into the wrong column within the row. Multiple update queries were used to update one or more columns at a time with strings read from columns containing the expected data.

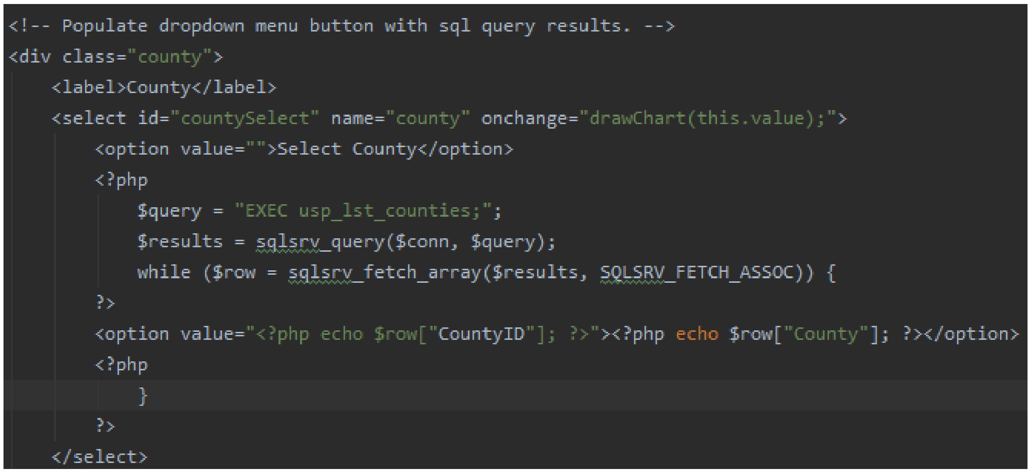
A future update will include a function to upload new files from the website and use the web server to parse fields by commas. If the first character of a field is a quotation mark, it would be expected that any following commas would be a part of the string until a closing quotation mark is found.

VISUALIZATION

The application uses the Google Charts API to display aggregated data to users in various forms, including column, bar, bubble, scatter, and line charts.

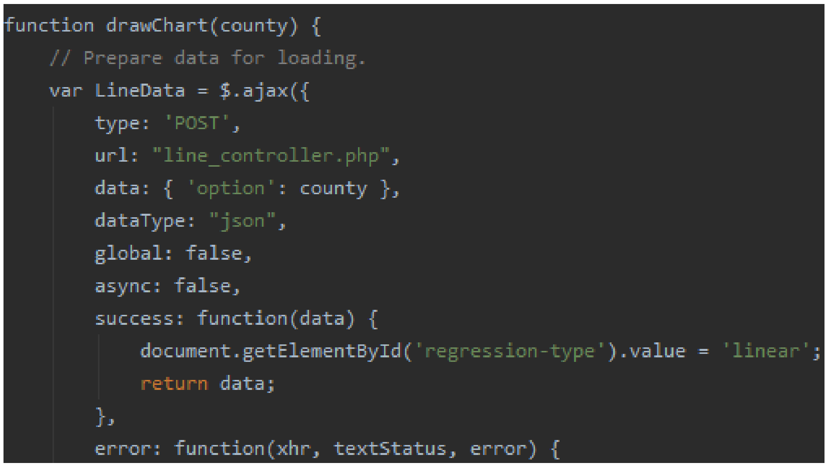
INTERACTIVE QUERIES

The application was developed using PHP server-side scripting to dynamically construct web pages upon client-request. Drop-down buttons query and populate their option values with record attributes from the database. Options with no associated records may be omitted.



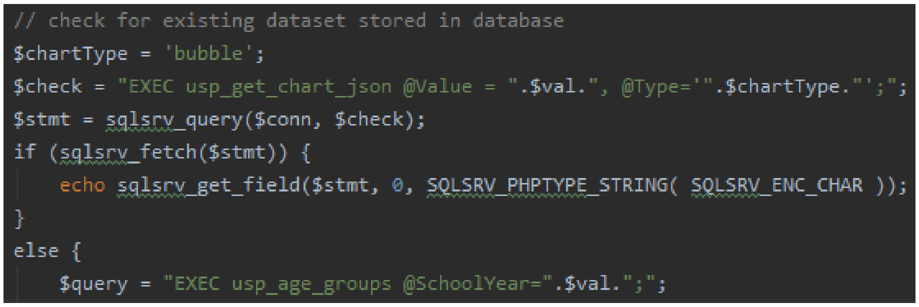
```
<!-- Populate dropdown menu button with sql query results. -->
<div class="county">
    <label>County</label>
    <select id="countySelect" name="county" onchange="drawChart(this.value);">
        <option value="">Select County</option>
        <?php
            $query = "EXEC usp_lst_counties;";
            $results = sqlsrv_query($conn, $query);
            while ($row = sqlsrv_fetch_array($results, SQLSRV_FETCH_ASSOC)) {
        ?>
        <option value="<?php echo $row["CountyID"]; ?>"><?php echo $row["County"]; ?></option>
        <?php
            }
        ?>
    </select>
```

Once a button value is selected, an AJAX call is made to update the chart's draw function. The function passes the selected value, a record index, to the controller script, which then queries a stored procedure using the value passed as a parameter. The resulting dataset is received by the controller and translated into a JSON string, which is then passed back into the view.



```
function drawChart(county) {
    // Prepare data for loading.
    var LineData = $.ajax({
        type: 'POST',
        url: "line_controller.php",
        data: { 'option': county },
        dataType: "json",
        global: false,
        async: false,
        success: function(data) {
            document.getElementById('regression-type').value = 'linear';
            return data;
        },
        error: function(xhr, textStatus, error) {
```

MACHINE-LEARNING

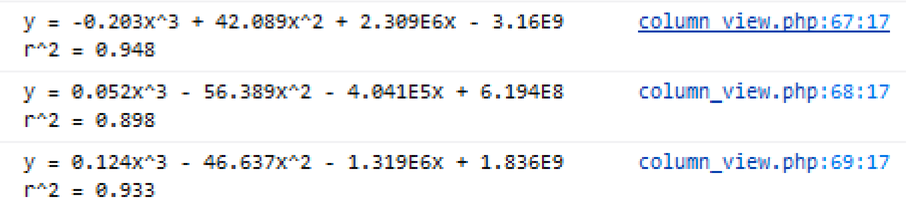


```
// check for existing dataset stored in database
$chartType = 'bubble';
$check = "EXEC usp_get_chart_json @Value = ".$val.", @Type='".$chartType."';";
$stmt = sqlsrv_query($conn, $check);
if (sqlsrv_fetch($stmt)) {
    echo sqlsrv_get_field($stmt, 0, SQLSRV_PHPTYPE_STRING( SQLSRV_ENC_CHAR ));
}
else {
    $query = "EXEC usp_age_groups @SchoolYear=".$val.";";
```

Supervised Learning is implemented within the application to reduce the computational load required by the RDBMS. Two variables, *Chart Type* and *User Selection* have been mapped as a primary key in a database table. If a pair's query has never been previously been processed, the application will proceed with querying the database in typical fashion; once the data is received and transformed into a JSON structure, the application will insert the JSON into the table along with the composite key. Future queries with the same composite key will result in an O(1) response as the JSON is queried from the hash-table. This benefit comes at the cost of increasing required storage, however the costs have been measured to be inconsequential in comparison to the performance benefits when simultaneous users are using the system.

ACCURACY

The coefficient of determination (R^2) is displayed with corresponding predictor functions in the web browser's console when loading line, column, or bar charts. R^2 has a range of [0, 1] and represents how well the predictor function represents the overall data set.



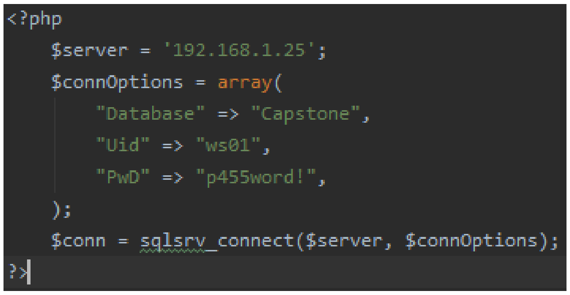
```
y = -0.203x^3 + 42.089x^2 + 2.309E6x - 3.16E9        column_view.php:67:17
r^2 = 0.948
y = 0.052x^3 - 56.389x^2 - 4.041E5x + 6.194E8        column_view.php:68:17
r^2 = 0.898
y = 0.124x^3 - 46.637x^2 - 1.319E6x + 1.836E9        column_view.php:69:17
r^2 = 0.933
```

The closer R^2 is to 1 indicates how close the predictor function fits to all plot points; however, a high R^2 value does not mean that predicted values will be accurate. Residual plots will need to be produced and visually assessed in order to determine whether a predictor function will be useful. Residuals are measured by subtracting expected results produced by the predictor function from the actual plotted values.

Due to the random nature of human behavior and performance, R^2 is expected to be quite low across the board. Analysis of residual plots may determine if a model is unbiased and supports the value of extrapolated information despite low R^2 values.

SECURITY AND MAINTENANCE

A service account queries the database using stored procedures that have been granted to the accounts permissions set. The service account is otherwise restricted from making any form of dynamic queries; this security measure was put in place to prevent unauthorized access to any tables in the database.



```php
<?php
    $server = '192.168.1.25';
    $connOptions = array(
        "Database" => "Capstone",
        "Uid" => "ws01",
        "PwD" => "p455word!",
    );
    $conn = sqlsrv_connect($server, $connOptions);
?>
```

For the purposes of this project, PII was scrubbed from the data set during the cleaning process; it's not expected of customers to do so with the product's deployment configuration: both web and database servers should be deployed on an internal network behind a firewall and not in a public-facing subnet, such as a DMZ. This configuration blocks external access to the product while allowing access to internal users.

Indexes were disabled in the web server's configuration files, blocking website directory access. Apache2 employs its own logging feature, tracking users access requests in logs include IP addresses, timestamps, and names of resources; these logs may be used to track and audit security concerns.

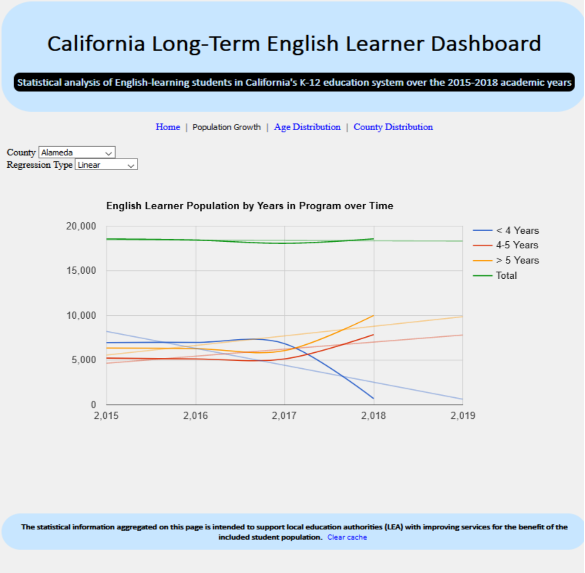
For development, testing, and maintenance purposes, the **php.ini** configuration file located in */etc/php/7.2/apache2/* can be adjusted to include or exclude classes of errors, including coding standards warnings. In a production environment, it's recommended that coding standards warnings are not collected in these server logs.

Should the product be adopted *en masse,* and a cloud solution be implemented, user accounts may be integrated into the system by storing authentication information within the database. A more realistic expectation would be that these applications would be appended to CALPADS.

INTERFACE

The product was fashioned with a minimalist design, limiting the content of each page to a single visualization of the data set. A navigation bar is centrally located below the header to draw attention to the information being presented.

A single iFrame is located at the center of the page and is used as the target for navigation links. Each chart page contains independent controls for variable selection.



California Long-Term English Learner Dashboard

Statistical analysis of English-learning students in California's K-12 education system over the 2015-2018 academic years

Home | Population Growth | Age Distribution | County Distribution

County: Alameda
Regression Type: Linear

English Learner Population by Years in Program over Time
20,000
15,000
10,000
5,000
0
2,015
2,016
2,017
2,018
2,019
< 4 Years
4-5 Years
> 5 Years
Total

The statistical information aggregated on this page is intended to support local education authorities (LEA) with improving services for the benefit of the included student population. Clear cache

SECTION D – DOCUMENTATION

BUSINESS REQUIREMENTS

Dynamic dataset

- Product is not constrained to displaying a static number of records; graphical representations of the data will adjust when changes are made to the dataset.

Designed for growth

- Product can expand to include new features without major reconstruction or requiring any amount of downtime during normal operational work hours.

Intuitive controls

- Product is easy to use and understand for non-technical users.
- Buttons have a clear purpose and do not require the need to refer to an instruction manual.

DATASET

Importing was done through SQL Server Management Studio's (SSMS) built-in tool to import records from a flat file into a single table. Unfortunately, school names containing commas caused several fields to be inserted into the wrong column within several thousand rows. Multiple update queries were used to update one or more columns at a time, setting fields to strings read from columns containing the correct data.

A future update will include the ability to upload files from the website and use the web server to parse fields by commas. If the first character of a field is a quotation mark, it would be expected that any following commas would be a part of the string until a closing quotation mark is found.

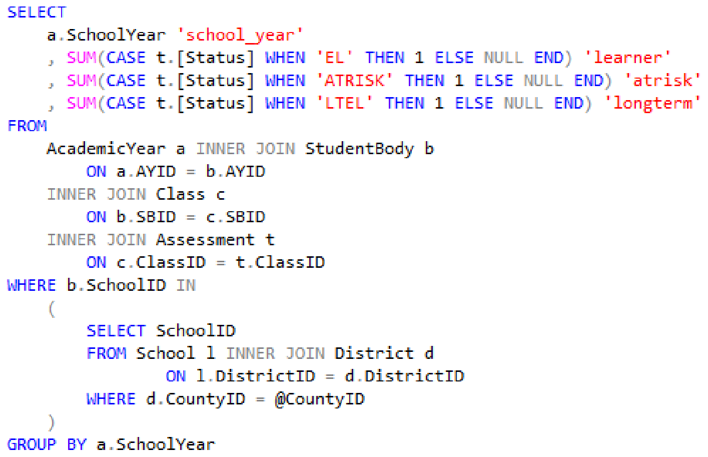
As an added measure to enforce referential integrity, two other files were collected from California's Department of Education; both files are available at https://www.cde.ca.gov/ds/si/ds/pubschls.asp. These files contain all schools and districts in the state, including state identifiers for both entities. Records from the student LTEL file were then associated to their respective schools through the school state identifiers.

Once the staging was completely cleaned, all business-data tables were populated.

DATA ANALYSIS CODE

New records were inserted into each table within the database using one or more fields from the staging table populated from the flat file import. Normalization of the database resulted in the creation of foreign key constraints for every table, so tables had to be loaded in order of dependency on parent tables.

Once all child tables were populated, a query was developed to determine the number of students in the state per year. Due to the volume of records being queried, selected columns were limited to those needed. This query formed the basis of the first chart to be used in the application, and a stored procedure was created to accept a parameter for variable selection determined by the user.

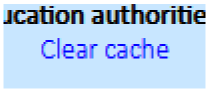


```
SELECT
    a.SchoolYear 'school_year'
    , SUM(CASE t.[Status] WHEN 'EL' THEN 1 ELSE NULL END) 'learner'
    , SUM(CASE t.[Status] WHEN 'ATRISK' THEN 1 ELSE NULL END) 'atrisk'
    , SUM(CASE t.[Status] WHEN 'LTEL' THEN 1 ELSE NULL END) 'longterm'
FROM
    AcademicYear a INNER JOIN StudentBody b
        ON a.AYID = b.AYID
    INNER JOIN Class c
        ON b.SBID = c.SBID
    INNER JOIN Assessment t
        ON c.ClassID = t.ClassID
WHERE b.SchoolID IN
    (
        SELECT SchoolID
        FROM School l INNER JOIN District d
                ON l.DistrictID = d.DistrictID
        WHERE d.CountyID = @CountyID
    )
GROUP BY a.SchoolYear
```

Similar queries were created following the same basic premise; depending on the application, different fields were used for variable selection.

HYPOTHESIS ACCEPTANCE ASSESSMENT

The product can accept new data and display the updated results, however the table storing JSON data from the machine-learning method will need to be manually deleted once new data is added. Since adding new data will be a process that only occurs once a year, this isn't a major issue, but a quality of life improvement for a future update would be to automatically empty the table once new records are inserted. A manual process exists to delete any existing records; this process is run by clicking **Clear cache** in the footer of the website.



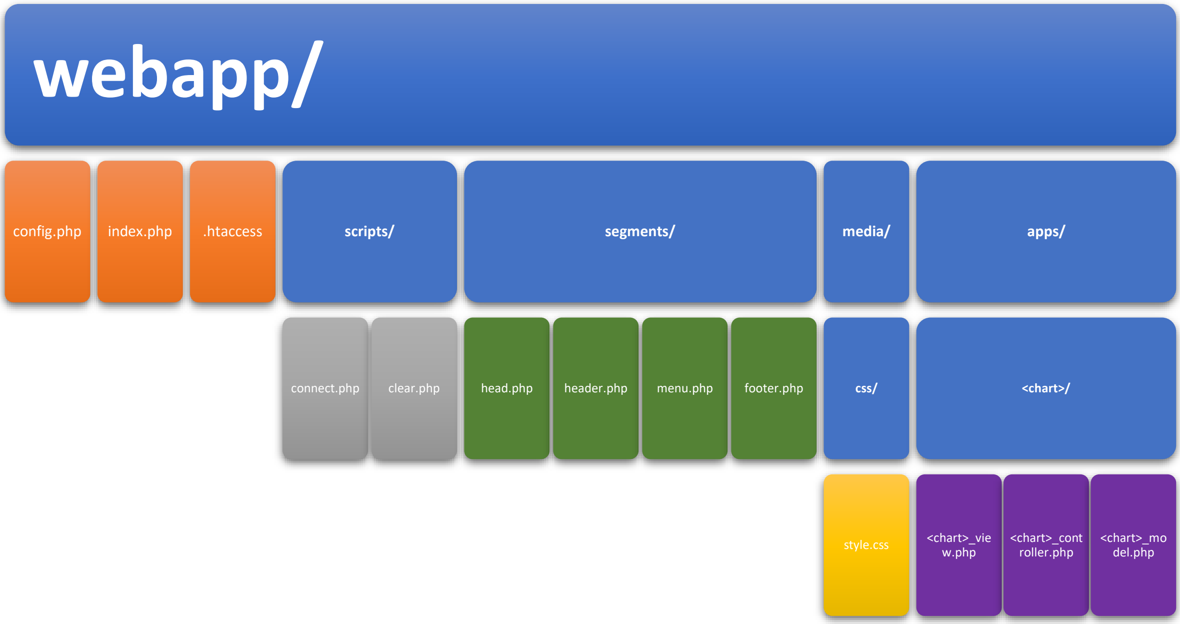
New charts were added to the product throughout the development process and were functional in shorter time than each of their predecessors. Charts can just as easily be adjusted by changing the data models and stored procedures used to query data.

The requirement that the product is accessible to non-technical users will significantly depend on customer feedback; usability testing will be conducted as an ongoing process throughout the life of the product using customer feedback to establish a metric for customer satisfaction.

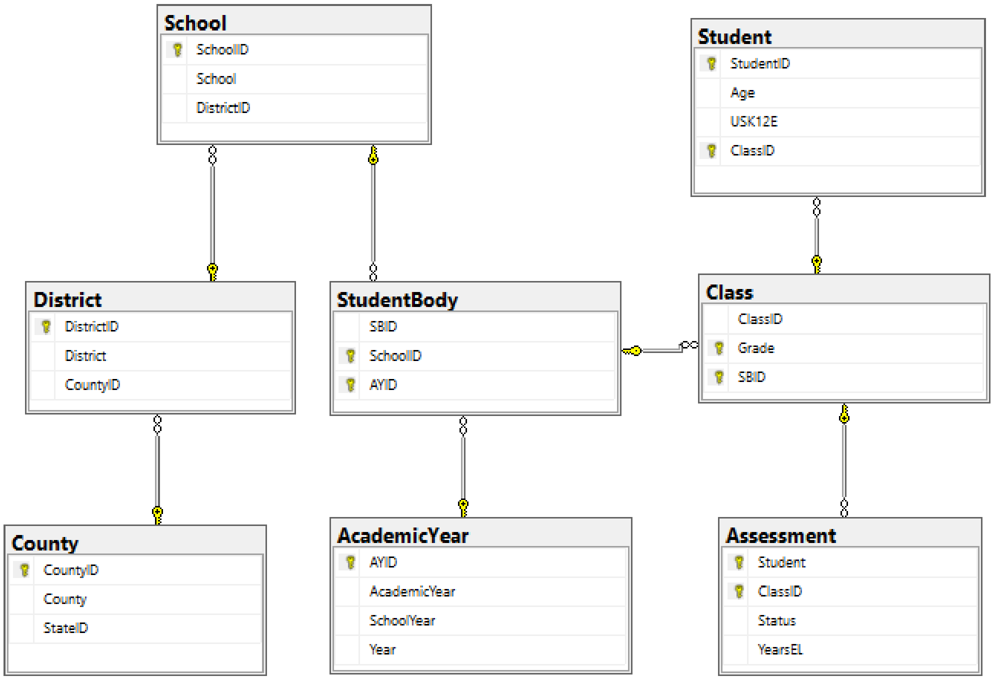
Barring non-functional requirements, the product is expected to pass acceptance testing.

STORYBOARD ELEMENTS

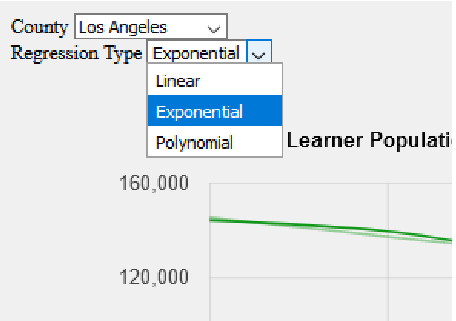
Web application directory structure and contents:



Database Table Diagram



ACCURACY ASSESSMENT AND TESTING RESULTS

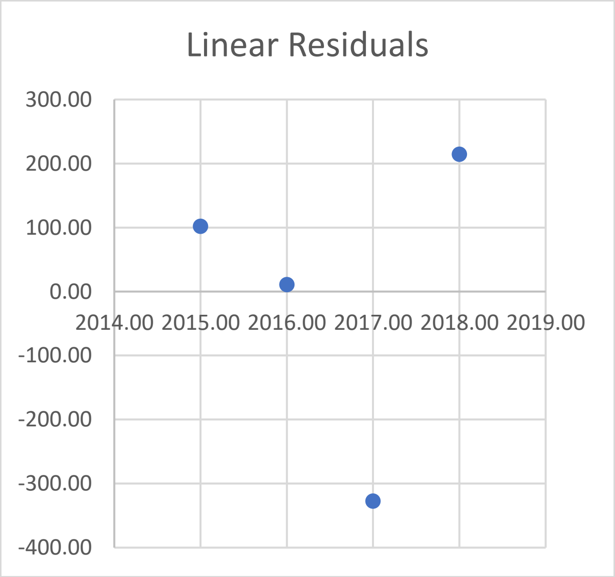

County Los Angeles
Regression Type Exponential

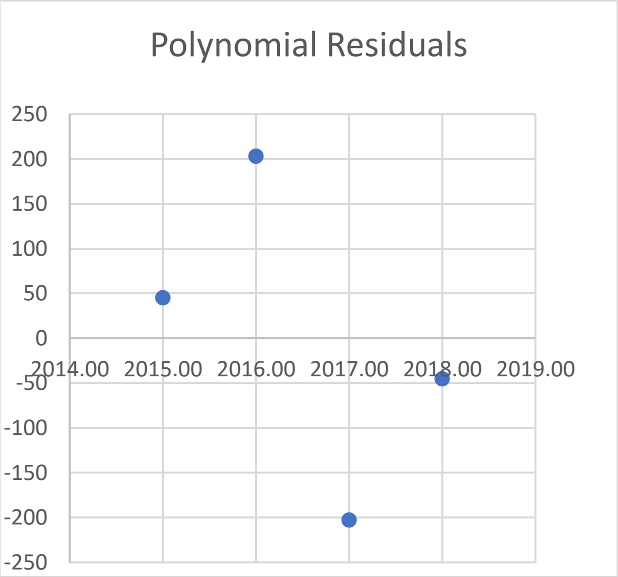
Line chart values were compared for a single county to determine if there were any discernable differences between residual plots using linear and polynomial regression methods. A button to toggle the selection of regression methods was added to the interface for line, bar, and column charts. A method was also added to the application to output regression functions and R^2 values to the web browser's console.

Alameda County Residual Results

Student Population	Linear R2	Polynomial R2
EL < 4 years	0.616	0.941
EL 4-5 years	0.573	0.936
EL > 5 years	0.539	0.912
Total	0.034	0.625

Year	Actual	Linear Expected	Linear Residual	Polynomial Expected	Polynomial Residual
2015.00	18557	18455.10	101.90	18511.691	45.309
2016.00	18432	18421.2	10.8	18228.99	203.01
2017.00	18060	18387.3	-327.3	18262.947	-202.947
2018.00	18568	18353.4	214.6	18613.372	-45.372





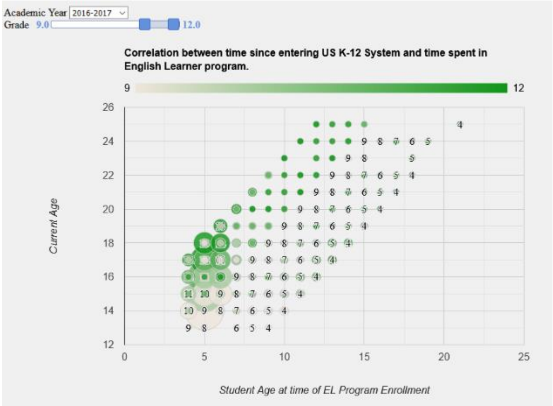
Polynomial R^2 was significantly higher across the board; residuals displayed no discernable pattern.

REVISIONS

The product uses the Google Charts API to produce graphical representations of data sets, as well as generate trendlines and coefficients of determination. A feature was planned to add residual charts to the site to assist with assessing the accuracy, however a dependency on an older version of the Charts API would have been required to do so. This reliance would've carried too much risk of losing support, so the feature was abandoned.

Initial efforts were focused on producing a functional prototype to allow usability testing to commence as soon as possible. Black-box component tests were conducted as each new module was produced to ensure the system was functioning as expected. Once a working prototype was ready, the product was ready to begin usability testing.

The product was well received in initial feedback in most areas, including learnability and efficiency. A bubble chart was included in the application, but the information displayed was confusing to most users and a replacement was deemed necessary. A scatter chart was the primary candidate, but delays in development due to differences in data structure blocked the update from this release candidate.



Academic Year 2016-2017
Grade 9.0 12.0

INSTALLATION

New Deployment - The following outline contains high-level information on deploying the web application within a local area network (LAN). Additional network configuration may be required to enable communication between hosts and clients.

1. Download & install Oracle's *Virtualbox* on the desired host machine.
    a. Note: Alternative hypervisors or dedicated hosts may also be used.
2. Download Ubuntu Server.
3. Create two new virtual machines using the Ubuntu Server image.
4. On the web server:
    a. Install *Apache2*.
    b. Install *PHP 7.2* or higher.
    c. Install *ODBC*.
    d. Install the *Microsoft SQL Server ODBC driver*.
5. On the database server:
    a. Install *Microsoft SQL Server*.
6. On a Windows client machine:

   a. Populate the data:
      i. Download and install *SQL Server Management Studio (SSMS).*
      ii. Run *SSMS* and execute the file **architecture.sql** as a query.
      iii. Run the powershell script **load.ps1** to populate the database tables.
   b. Deploy the website:
      i. Download and install *JetBrain's PyCharm Professional Edition*.
      ii. Run PyCharm and open the **webapp** directory as a project.
      iii. Click **Tools** -> **Deployment** -> **Configuration** and create a connection to allow file uploads to the web server using SFTP.
      iv. Upload the project files to **/var/www/html/** on the web server.
7. In a web browser, navigate to the web server's address to verify the application is available.

Redeployment – Using existing virtual machine images provided.

1. Download and install Oracle's Virtualbox on the desired host machine.
2. In Virtualbox:
   a. Click New, enter a name for the web server, and select Use an existing virtual hard disk file.
      - Navigate to and select server1.vdi in the included files.
      - Follow the directions in the guided wizard to finish deployment of the web server.
   b. Click New, enter a name for the SQL server, and select Use an existing virtual hard disk file.
      - Navigate to and select sql_server.vdi in the included files.
      - Follow the directions in the guided wizard to finish deployment of the SQL server.
3. Run both SQL and web servers.
4. Open a web browser and navigate to ws01.local
   - You may have to enter ws01.local:8000 or <web server IP address>:8000

SECTION E – SOURCES

1. **California Department of Education**; January 29, 2019. Available at
   https://www.cde.ca.gov/sp/me/mt/programs.asp
   Accessed September 14, 2019.
2. **Google**; August 31, 2018. Available at
   https://developers.google.com/chart/interactive/docs/reference#dataparam
   Accessed September 14, 2019.
3. **U.S. Department of Education**; September 20, 2019. Available at
   https://studentprivacy.ed.gov/content/personally-identifiable-information-pii
   Accessed September 20, 2019.
4. **Rooker, Leroy S.** (2004, 18 November)
   Letter re: Disclosure of Anonymous Data under FERPA
   Retrieved from https://www2.ed.gov/policy/gen/guid/fpco/ferpa/library/nashville_tn2004.html
5. **Californa School Dashboard**; September 19, 2019. Available at
   https://www.caschooldashboard.org/reports/42693100000000/2018
   Accessed September 19, 2019.