Workforce Allocation Optimization Tool

Technical Specifications

05/25/2018

Public Health Informatics Institute / HITRAC

|  |  |
| --- | --- |
| Last Modified Date | 05/25/2018 |
| Version | 2.2 |
| Document Title | Allocation Tool Technical Specification |
| Author(s) | Takunda Dhlakama, Lynnette Musekiwa and Morris Baradza |
| Original Authors | Emily Gooding and Colleen Gootee (Georgia Institute of Technology) |
| Document Status | Draft 2 |
| Reviewers | Keith Waters and Swati Sharma (CDC), and Juneka Rembert (PHII) |
| Revision Reasons | The Workforce Allocation Optimization Tool was modified from an Excel tool to a web-based tool. This document was updated to reflect the revisions. |

**Contents**

[Acknowledgments 4](#_Toc515024881)

[Workforce Allocation Optimization Tool overview 5](#_Toc515024882)

[Optimization model overview 6](#_Toc515024883)

[Mixed integer program description 6](#_Toc515024884)

[Model’s logic short examples 11](#_Toc515024885)

[Implementation in GLPK 20](#_Toc515024886)

[Tool interface description 21](#_Toc515024887)

[Accessing the tool 21](#_Toc515024888)

[Uploading Data 21](#_Toc515024889)

[Maintaining data integrity 21](#_Toc515024890)

[Timeframe consistency 21](#_Toc515024891)

[Clearing input and results 22](#_Toc515024892)

[System Configuration 22](#_Toc515024893)

[General Settings 23](#_Toc515024894)

[Worker Types - In the database, this table is named “worker\_type.” 24](#_Toc515024895)

[Worker Levels - In the database, this table is named “worker\_level.” 24](#_Toc515024896)

[Worker Salaries- In the database, this table is named “worker\_salary.” 25](#_Toc515024897)

[Demand Locations - In the database, this table is named “demand\_location.” 25](#_Toc515024898)

[Study Locations - In the database, this table is named “location.” 26](#_Toc515024899)

[Distances Lookup Table - In the database, this table is named “distance.” 27](#_Toc515024900)

[Worker Settings - In the database, this table is named “graduate.” 27](#_Toc515024901)

[Fixed Assignments 31](#_Toc515024902)

[Workers Assignments 32](#_Toc515024903)

[Demand Met by Location 33](#_Toc515024904)

[Assignments by Location 33](#_Toc515024905)

[Assignments by Worker Types 34](#_Toc515024906)

[Preferences by Location 34](#_Toc515024907)

[Running the model 36](#_Toc515024908)

[Data files 36](#_Toc515024909)

[demand.csv 37](#_Toc515024910)

[general.csv 37](#_Toc515024911)

[locations.csv 37](#_Toc515024912)

[preferences.csv 38](#_Toc515024913)

[Types.csv 39](#_Toc515024914)

[Workers.csv 39](#_Toc515024915)

[GLPK Model 40](#_Toc515024916)

[Recommend Security Features 43](#_Toc515024917)

[Physical Access 43](#_Toc515024918)

[Application/Software Level 43](#_Toc515024919)

[System Installation 44](#_Toc515024920)

[Installation on a Linux Ubuntu Server 44](#_Toc515024921)

[Installation on a Windows Server 45](#_Toc515024922)

[“ModelDB.txt” 55](#_Toc515024923)

# Acknowledgments

The Workforce Allocation Optimization Tool was modeled from work rendered by the Public Health Informatics Institute (PHII) on the Centers for Disease Control and Prevention (CDC) funded Mozambique Workforce Planning and Deployment Optimization project. The Workforce Allocation Optimization Tool was initially developed by students Emily Gooding and Colleen Gootee under the advice of Pinar Keskinocak and Julie Swann from the Georgia Institute of Technology, as a project for their senior design class, and in collaboration with PHII (September 2013 – June 2014). Enhancements on the tool’s optimization model, the user interface and documentation were developed by independent contractor Monica Villarreal (August 2014 – October 2014), in consultation with PHII, CDC HQ Atlanta, CDC Mozambique and Jhpiego. The tool was developed to strengthen the human resources for health workforce allocation assignment process.

The Public Health Informatics Institute would like to thank the Georgia Institute of Technology for the initial development of the Workforce Allocation Optimization Tool. HITRAC has taken up the Excel tool and transformed the Workforce Allocation Optimization Tool into a distributed web system. HITRAC has maintained how the tool runs and has added a web interface to capture input and display Workforce Allocation Optimization Tool results.

# Workforce Allocation Optimization Tool overview

The objective of the Workforce Allocation Optimization Tool is to support the assignment of recent healthcare graduates’, i.e., future healthcare workers (HCWs), based on the demand of different locations, the HCWs’ salaries and potential budget restrictions, and the HCWs’ location preferences for assignment, with the following goals: (i) to fulfill each location’s demand, as possible, and (ii) to maximize the HCWs’ satisfaction resulting from the assignments.

We consider the **percentage of fulfilled** **demand** as the metric of interest regarding a location’s demand fulfillment, with the goal of obtaining a fairer demand assignment. For instance, it is not the same to just fulfill one out of four nurses demand, i.e., 25% of nurses demand, in location A (a deficit of three workers), than to fulfill seven out of 10 nurses demand, i.e., 70% of nurses demand, in location B (also a deficit of three workers).

HCWs rank a few locations as their preferred location assignments (usually three locations). We consider a **preference score** **(a weight for the HCW assignment to the preferred location)** for each ranked location as a metric of the HCW satisfaction given the assignment. We consider that by assigning HCWs to one of their preferred locations, we are rewarded with the corresponding preference (i.e., satisfaction) score.

The tool has three main elements:

1. The web Workforce Allocation Optimization Tool, referred as “tool” in this document, provides an interface for the input data, generates the input data files for the optimization solver, and reads and provides the solver’s output as well as relevant statistics for analysis.
2. The GLPK package (“glpsol.exe”, “glpk\_4\_57.dll”) contains the open-license optimization solver. GLPK is a free alternative to commercial optimization solvers. This package runs in Windows and Linux.
3. The GLPK optimization model (“Model.txt” and “ModelDB.txt”) contains the implementation of the optimization model in a language familiar to GLPK (MathProg). The model includes instructions about the input and output data files, in addition to the optimization model, which makes decisions regarding the HCWs’ assignments considering the input data and the model constraints.

# Optimization model overview

The objective of the model is to maximize (i.e., choosing the best among the feasible choices of HCW assignments) the **total** **rewards (preference scores or weights)** coming from assigning workers to their preferred locations **minus** the **penalties** that result from **not fulfilling a percentage of the locations’ demand for each worker type**, subject to the following constraints:

* Each worker can be assigned to at most one location.
* The allocation process cannot assign more workers than those demanded by the location.
* The allocation process cannot violate any budget constraints (if there are any budget constraints).
* Fixed workers must be assigned to their fixed location.
* Some workers may only be assigned to one of their preferences (or not be assigned at all).

To obtain the best solution (or one of the best solutions if there are more than one) for this problem, we formulated an optimization model, particularly a mixed integer program (MIP), which can be solved by an optimization solver (such as GLPK). Linear programming (LP) is a technique to optimize a linear objective function, subject to linear equalities and inequalities constraints. An MIP is an LP that includes variables that are not continuous (such as variables that can only take integer values or zero/one values).

## Mixed integer program description

The content described next is highly technical. Users do not need to understand the equations in detail and can skip to the Model’s Logic Short Examples section.

To describe an MIP we need to declare:

1. Sets: the model’s collections of objects (locations, workers, etc.).
2. Parameters: describe the characteristics of the model.
3. Variables: represent each of the decisions that the model needs to make.
4. Objective function (OF): a linear function that the model seeks to maximize (or minimize) given the decisions it makes.
5. Constraints: sets of linear restrictions over the decisions that can be made.

Below, we list each of these five elements for the proposed MIP. The left column includes the mathematical symbols and expressions, and the right column describes them.

|  |  |
| --- | --- |
| Sets | Description |
|  | Set of demand locations. |
|  | Set of worker types (worker categories). |
|  | Set of workers (HCWs) to assign. |
|  | Set of pairs of workers and preferred locations . |

|  |  |
| --- | --- |
| Parameters | Description |
|  | Total budget to cover salaries. If there is not a budget, then it is equal to “NA” (not available). |
|  | Budget to cover salaries for location. If there is not a budget for a given location, then it is equal to “NA” (not available). |
|  | Penalty for the fraction of unfulfilled demand for location . |
|  | Type of worker . |
|  | Salary of worker . |
|  | Location to where worker can be pre-fixed (pre-assigned). |
|  | =1 if worker is pre-fixed to location , =0 otherwise. |
|  | =1 if worker can only be assigned to a location of his/her preference, where , =0 otherwise. |
|  | Demand for location for each worker type in number of workers per type. |
|  | Preference score for assigning worker to a location of his/her preference, where . |

|  |  |
| --- | --- |
| Variables | Description |
|  | =1 if worker is assigned to location and =0 otherwise (i.e., these are binary variables), for all and such that one of these three conditions hold:  1) and (i.e., if the worker is pre-fixed to a location, we just create the assignment variable for such location), or  2) , , and (i.e., if the worker can only be assigned to one of his/her preferences, we just create the assignment variables for those preferred locations), or  3) and (i.e, if worker can be assigned to any location, we create an assignment variable for all locations ). |
|  | Fraction of fulfilled demand for location for worker type (these variables are greater than zero). |

|  |  |
| --- | --- |
| Objective Function | Description |
|  | Maximize the sum of the preference scores or weights for assigning workers to their preferred locations, minus the penalties for unfulfilled fraction of demands. |

|  |  |
| --- | --- |
| Constraints | Description |
|  | The pre-fixed workers are assigned to their fixed locations. |
|  | The fraction of demand cannot be greater than 1 (it ranges from 0 to 1) |
|  | Each workercan be assigned to at most one location. Note that the summation covers all and only the variables that were created as described above (conditions 1), 2) or 3)). |
|  | The salaries of all the workers assigned to each location cannot exceed the location’s budget. Note that if there is not such budget (i.e.,), this constraint is not considered for that location. |
|  | The salaries of all the workers assigned to all locations cannot exceed the total budget. Note that if there is not such budget (i.e.,), this constraint is not considered. |
|  | The number of assigned workers of a type cannot exceed the location’s demand . Note how in the summation only workers of type are added. |
|  | The number of assigned workers of a type covers a fraction of the location’s demand. |

## Model’s logic short examples

For someone that is not familiar with LPs and MIPs or with optimization models in general, it would be harder to understand the logic behind the proposed optimization model described above. We created some simple examples that illustrate the decisions that would be taken by the model, considering the given assumptions.

**Example 1**

Assumptions:

* There is one HCW type.
* There are three locations: L1, L2, and L3.
* Each location demands one worker.
* The penalty for unfulfilled demand percentage is 40 (for all locations).
* There are three HCWs: W1, W2, and W3. They have the following preferences and preference scores (weights):

|  |  |  |  |
| --- | --- | --- | --- |
| HCW | Preference 1, Weight 1 | Preference 2,  Weight 2 | Preference 3,  Weight 3 |
| W1 | L1,3 | L2,2 | L3,1 |
| W2 | L1,3 | L1,2 | L2,1 |
| W3 | L2,3 | L1,2 | L3,1 |

Note that W2 selected L1 as both his first and second preferred location. For each location, we only consider the weight of the highest rank. In this case, if we assign W2 to L1, we get a reward of three (and not 3 + 2 = 5). If we assign W2 to L2, we get a reward of one, and we get no rewards for an assignment to L3.

We can list all the potential assignments for the three workers (an “X” means that the HCW was not assigned to any location):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | W1 | W2 | W3 | OF Calculation | OF Value |
| 1 | X | X | X | -40(3) | -120 |
| 2 | L1 | X | X | -40(2) + 3 | -77 |
| 3 | L2 | X | X | -40(2) + 2 | -78 |
| 4 | L3 | X | X | -40(2) + 1 | -79 |
| 5 | X | L1 | X | -40(2) + 3 | -77 |
| 6 | X | L2 | X | -40(2) + 1 | -79 |
| 7 | X | L3 | X | -40(2) + 0 | -80 |
| 8 | X | X | L1 | -40(2) + 2 | -78 |
| 9 | X | X | L2 | -40(2) + 3 | -77 |
| 10 | X | X | L3 | -40(2) + 1 | -79 |
| 11 | L1 | L2 | X | -40(1) + 3 + 1 | -36 |
| 12 | L1 | L3 | X | -40(1) + 3 + 0 | -37 |
| 14 | L2 | L3 | X | -40(1) + 2 + 0 | -38 |
| 15 | L3 | L1 | X | -40(1) + 1 + 3 | -36 |
| 16 | L3 | L2 | X | -40(1) + 1 + 1 | -38 |
| 17 | L1 | X | L2 | -40(1) + 3 + 3 | -34 |
| 18 | L1 | X | L3 | -40(1) + 3 + 1 | -36 |
| 19 | L2 | X | L1 | -40(1) + 2 + 2 | -36 |
| 20 | L2 | X | L3 | -40(1) + 2 + 1 | -37 |
| 21 | L3 | X | L1 | -40(1) + 1 + 2 | -37 |
| 22 | L3 | X | L2 | -40(1) + 1 + 3 | -36 |
| 23 | X | L1 | L2 | -40(1) + 3 + 3 | -34 |
| 24 | X | L1 | L3 | -40(1) + 3 + 1 | -36 |
| 25 | X | L2 | L1 | -40(1) + 1 + 2 | -37 |
| 26 | X | L2 | L3 | -40(1) + 1 + 1 | -38 |
| 27 | X | L3 | L1 | -40(1) + 0 + 2 | -38 |
| 28 | X | L3 | L2 | -40(1) + 0 + 3 | -37 |
| 29 | L1 | L2 | L3 | 3 + 1 + 1 | 5 |
| 30 | L1 | L3 | L2 | 3 + 0 + 3 | 6 |
| 31 | L2 | L1 | L3 | 2 + 3 + 1 | 6 |
| 32 | L2 | L3 | L1 | 2 + 0 + 2 | 4 |
| *33* | ***L3*** | ***L1*** | ***L2*** | ***1 + 3 + 3*** | ***7*** |
| 34 | L3 | L2 | L1 | 1 + 1 + 2 | 4 |

Under the given assumptions, there are 34 different ways (or feasible solutions) to assign the three workers. For instance, consider the solution #25. L1 and L2 have 100% of their demand fulfilled, and L3 has 0%. This means that L3 has 100% unfilled demand, which corresponds to a penalty of -40(1) = -40. Then there is a reward of one because W2 is assigned to his third preference L2 (with a weight of 1), and a reward of two because worker W3 is assigned to his second preference L1 (with a weight of 2). We add the penalties and rewards (preference scores), and we get -40 + 1 + 2 = -37. We compute the objective function value for all the other feasible solutions. The solution with the largest objective value is #33, with a value of seven. This means that the model will recommend, among all possible solutions, to assign W1 to L3, W2 to L1 and W3 to L2.

**Example 2**

Assumptions:

* There is one HCW type.
* There are three locations: L1, L2 and L3.
* Each location demands one worker.
* The penalty for unfulfilled demand percentage is 40 (for all locations).
* **Budgets are 30,000; 30,000 and 25,000 for locations L1, L1 and L3 respectively.**
* There are three HCWs: W1, W2 and W3. They have the following salaries, preferences and preference scores (weights):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HCW | Salary | Preference 1, Weight 1 | Preference 2,  Weight 2 | Preference 3,  Weight 3 |
| W1 | 30,000 | L1,3 | L2,2 | L3,1 |
| W2 | 25,000 | L1,3 | L1,2 | L2,1 |
| W3 | 25,000 | L2,3 | L1,2 | L3,1 |

This example is similar to Example 1, with the difference of the added budget constraints. Since L3 has a budget constraint of 25,000 and the salary of W1 is 30,000, any solution that involves assigning W1 to L3 is no longer feasible. We repeated the process above, enumerating all the potential solutions, and we changed the color for those now unfeasible solutions to gray.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | W1 | W2 | W3 | OF Calculation | OF Value |
| 1 | X | X | X | -40(3) | -120 |
| 2 | L1 | X | X | -40(2) + 3 | -77 |
| 3 | L2 | X | X | -40(2) + 2 | -78 |
| 4 | L3 | X | X | -40(2) + 1 | -79 |
| 5 | X | L1 | X | -40(2) + 3 | -77 |
| 6 | X | L2 | X | -40(2) + 1 | -79 |
| 7 | X | L3 | X | -40(2) + 0 | -80 |
| 8 | X | X | L1 | -40(2) + 2 | -78 |
| 9 | X | X | L2 | -40(2) + 3 | -77 |
| 10 | X | X | L3 | -40(2) + 1 | -79 |
| 11 | L1 | L2 | X | -40(1) + 3 + 1 | -36 |
| 12 | L1 | L3 | X | -40(1) + 3 + 0 | -37 |
| 14 | L2 | L3 | X | -40(1) + 2 + 0 | -38 |
| 15 | L3 | L1 | X | -40(1) + 1 + 3 | -36 |
| 16 | L3 | L2 | X | -40(1) + 1 + 1 | -38 |
| 17 | L1 | X | L2 | -40(1) + 3 + 3 | -34 |
| 18 | L1 | X | L3 | -40(1) + 3 + 1 | -36 |
| 19 | L2 | X | L1 | -40(1) + 2 + 2 | -36 |
| 20 | L2 | X | L3 | -40(1) + 2 + 1 | -37 |
| 21 | L3 | X | L1 | -40(1) + 1 + 2 | -37 |
| 22 | L3 | X | L2 | -40(1) + 1 + 3 | -36 |
| 23 | X | L1 | L2 | -40(1) + 3 + 3 | -34 |
| 24 | X | L1 | L3 | -40(1) + 3 + 1 | -36 |
| 25 | X | L2 | L1 | -40(1) + 1 + 2 | -37 |
| 26 | X | L2 | L3 | -40(1) + 1 + 1 | -38 |
| 27 | X | L3 | L1 | -40(1) + 0 + 2 | -38 |
| 28 | X | L3 | L2 | -40(1) + 0 + 3 | -37 |
| 29 | L1 | L2 | L3 | 3 + 1 + 1 | 5 |
| *30* | ***L1*** | ***L3*** | ***L2*** | ***3 + 0 + 3*** | ***6*** |
| *31* | ***L2*** | ***L1*** | ***L3*** | ***2 + 3 + 1*** | ***6*** |
| 32 | L2 | L3 | L1 | 2 + 0 + 2 | 4 |
| 33 | L3 | L1 | L2 | 1 + 3 + 3 | 7 |
| 34 | L3 | L2 | L1 | 1 + 1 + 2 | 4 |

From the table above, we know that both solutions #30 and #31 have the same OF value of 6. There are two optimal solutions for this example, and GLPK will report which ever solution it finds first. **If nothing changes in the data input (including the order of the data), GLPK should find the same solution every time the model is run.**

**Example 3**

Assumptions:

* There is one HCW type.
* There are three locations: L1, L2 and L3.
* Each location demands one worker.
* The penalty for unfulfilled demand percentage is 40 (for all locations).
* There are three HCWs: W1, W2 and W3. **W1 is fixed to L1**. They have the following preferences and preference scores (weights):

|  |  |  |  |
| --- | --- | --- | --- |
| HCW | Preference 1, Weight 1 | Preference 2,  Weight 2 | Preference 3,  Weight 3 |
| W1 | L1,3 | L2,2 | L3,1 |
| W2 | L1,3 | L1,2 | L2,1 |
| W3 | L2,3 | L1,2 | L3,1 |

This example is similar to Example 1, with the difference that W1 is fixed to L1. Thus, any solution that does not include W1 assigned to L1 is not feasible anymore. Again, we enumerated all the potential solutions, and we changed the color to gray for those solutions now unfeasible.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | W1 | W2 | W3 | OF Calculation | OF Value |
| 1 | X | X | X | -40(3) | -120 |
| 2 | L1 | X | X | -40(2) + 3 | -77 |
| 3 | L2 | X | X | -40(2) + 2 | -78 |
| 4 | L3 | X | X | -40(2) + 1 | -79 |
| 5 | X | L1 | X | -40(2) + 3 | -77 |
| 6 | X | L2 | X | -40(2) + 1 | -79 |
| 7 | X | L3 | X | -40(2) + 0 | -80 |
| 8 | X | X | L1 | -40(2) + 2 | -78 |
| 9 | X | X | L2 | -40(2) + 3 | -77 |
| 10 | X | X | L3 | -40(2) + 1 | -79 |
| 11 | L1 | L2 | X | -40(1) + 3 + 1 | -36 |
| 12 | L1 | L3 | X | -40(1) + 3 + 0 | -37 |
| 14 | L2 | L3 | X | -40(1) + 2 + 0 | -38 |
| 15 | L3 | L1 | X | -40(1) + 1 + 3 | -36 |
| 16 | L3 | L2 | X | -40(1) + 1 + 1 | -38 |
| 17 | L1 | X | L2 | -40(1) + 3 + 3 | -34 |
| 18 | L1 | X | L3 | -40(1) + 3 + 1 | -36 |
| 19 | L2 | X | L1 | -40(1) + 2 + 2 | -36 |
| 20 | L2 | X | L3 | -40(1) + 2 + 1 | -37 |
| 21 | L3 | X | L1 | -40(1) + 1 + 2 | -37 |
| 22 | L3 | X | L2 | -40(1) + 1 + 3 | -36 |
| 23 | X | L1 | L2 | -40(1) + 3 + 3 | -34 |
| 24 | X | L1 | L3 | -40(1) + 3 + 1 | -36 |
| 25 | X | L2 | L1 | -40(1) + 1 + 2 | -37 |
| 26 | X | L2 | L3 | -40(1) + 1 + 1 | -38 |
| 27 | X | L3 | L1 | -40(1) + 0 + 2 | -38 |
| 28 | X | L3 | L2 | -40(1) + 0 + 3 | -37 |
| 29 | L1 | L2 | L3 | 3 + 1 + 1 | 5 |
| *30* | ***L1*** | ***L3*** | ***L2*** | ***3 + 0 + 3*** | ***6*** |
| 31 | L2 | L1 | L3 | 2 + 3 + 1 | 6 |
| 32 | L2 | L3 | L1 | 2 + 0 + 2 | 4 |
| 33 | L3 | L1 | L2 | 1 + 3 + 3 | 7 |
| 34 | L3 | L2 | L1 | 1 + 1 + 2 | 4 |

From the table above, we know that solution #30 has the largest OF value.

### 

**Example 4**

Assumptions:

* There is one HCW type.
* There are two locations: L1 and L2. L1 needs two workers and L2 needs one.
* The penalty for unfulfilled demand percentage is 40 (for all locations).
* There are two HCWs: W1 and W2. They have the following preferences and preference scores (weights):

|  |  |  |
| --- | --- | --- |
| HCW | Preference 1, Weight 1 | Preference 2,  Weight 2 |
| W1 | L1,3 | L2,2 |
| W2 | L1,3 | L1,2 |

We enumerate the potential solutions:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | W1 | W2 | OF Calculation | OF Value |
| 1 | X | X | -40(2) | -80 |
| 2 | L1 | X | -40(.5) – 40(1) + 3 | -57 |
| 3 | L2 | X | -40(1) + 2 | -38 |
| 4 | X | L1 | -40(.5) – 40(1) + 3 | -57 |
| 5 | X | L2 | -40(1) + 0 | -40 |
| 6 | L1 | L1 | -40(1) + 3 + 3 | -34 |
| 7 | L1 | L2 | -40(.5) + 3 + 0 | -17 |
| ***8*** | ***L2*** | ***L1*** | ***-40(.5) + 2 + 3*** | ***-15*** |

The best solution for this example is #8, with an OF value of -15. W1 is assigned to L2, so we get a reward of two (as it was his second choice), and W2 is assigned to L1, so we get a reward of three (his first choice). Note that there is one worker, out of a demand of two, which is assigned to L1, so 50% of L1 demand is fulfilled. This is why we have a penalty of -40(.50) = -20. The demand for L2 is totally fulfilled (with one worker assigned), so there is not an unfulfilled demand penalty associated with L2. **The fact that the unfilled demand is penalized by the percentage and not the total number of workers allows for a fairer distribution of workers among different locations of different demand sizes**.

### 

**Example 5**

Assumptions:

* There is one HCW type.
* There are two locations: L1 and L2. L1 needs two workers and L2 needs one.
* **The penalty for unfulfilled percentage of demand is 80 for L1 and 40 for L2.**
* There are two HCWs: W1 and W2. They have the following preferences and preference scores (weights):

|  |  |  |
| --- | --- | --- |
| HCW | Preference 1, Weight 1 | Preference 2,  Weight 2 |
| W1 | L1,3 | L2,2 |
| W2 | L1,3 | L1,2 |

We enumerate the potential solutions:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | W1 | W2 | OF Calculation | OF Value |
| 1 | X | X | -80(1) – 40(1) | -120 |
| 2 | L1 | X | -80(.5) – 40(1) + 3 | -77 |
| 3 | L2 | X | -80(1) + 2 | -78 |
| 4 | X | L1 | -80(.5) – 40(1) + 3 | -77 |
| 5 | X | L2 | -80(1) + 0 | -80 |
| ***6*** | ***L1*** | ***L1*** | ***-40(1) + 3 + 3*** | ***-34*** |
| 7 | L1 | L2 | -80(.5) + 3 + 0 | -37 |
| 8 | L2 | L1 | -80(.5) + 2 + 3 | -35 |

The best solution for this example is #6. Both W1 and W2 are assigned to L1, their first choices, so we get a reward of six. Also, L1 demand is totally fulfilled, so there is not an unfulfilled penalty associated with L1. No demand of L2 is fulfilled, so there is a penalty of 40. This shows how we can use the unfulfilled demand percentage penalty to give more weight to some locations that might be more important.

## 

## Implementation in GLPK

The optimization model described previously was implemented in GLPK. The GLPK (GNU Linear Programming Kit) package is free software intended for solving large-scale LPs, MIPs and other related problems. GLPK supports the GNU MathProg modeling language. For more information: <https://www.gnu.org/software/glpk/>.

The implementation of the model can be found in the file “Model.txt”. In the first section of the file, all the sets and parameters are declared and instructions are given so that the data is read from the input files. Then, the decision variables are declared. The objective function is declared next, followed by the different model’s constraints (expressed in terms of the declared sets, parameters and variables). Finally, there are instructions to create output files with the model’s results.

From the model’s logic examples, note that **GLPK’s solution algorithm to solve the optimization model does not just follow a sequential algorithm** to assign the HCWs**. It actually considers all the feasible solutions**. However, most of the cases GLPK will not have to test every single solution (as there will be exponentially many and it could take too long to do so), because the solution algorithm implemented in GLPK moves toward what looks like a good solution. Also, the solution algorithm in GLPK stops once it has proven that the best solution on-hand is not very far from the best solution possible (with a gap no larger than the minimum gap). The math behind this stop criterion is not straightforward, but quite complex solution algorithms using optimization theory have been implemented in optimization solvers like GLPK.

**If there are two or more optimal solutions, GLPK will report whichever solution it finds first.** **If nothing changes in the data input (including the order of the data), GLPK should find the same solution every time the model is run.**

# Tool interface description

The new Workforce Allocation Optimization Tool is the web-based adaptation of the Excel allocation tool (referenced in the Acknowledgements section above). It was developed using CodeIgniter PHP framework and MySQL Database. The Web version runs and generates csv files which are later read back in to the system. All the input and output data are directly saved in the database. The system runs the GLPK solver as an independent process to generate outputs. These outputs are further used to generate required reports.

## Accessing the tool

The Workforce Allocation Optimization Tool runs on Windows and Linux. Minimum server specifications are Server Intel Core i5 of 2.3 GHZ and recommended minimum RAM 4GB.

## Uploading Data

For easier integration with other systems, the system has a csv file upload template. Data is imported from left to right, i.e. first column in imported file would match first column in the system.

## Maintaining data integrity

**To maintain data integrity, most of the input screens have validation rules that display error messages on violation.** For instance, mandatory fields are marked with a red asterisk, and duplicates will be flagged whenever possible. In addition, some fields have minimum and/or maximum potential values like salaries and budgets that cannot be negative. For instance, required fields are marked with a red asterisk. Duplicates will be flagged whenever possible.

**If the data are prepared outside the tool and uploaded, values in the import values must match elements in the system to be correctly updated.** For example, when uploading data from a csv import, the preference location has to match a saved demand location name. Otherwise, the data do not get uploaded,or therewill be problems running the model in GLPK and showing the results.

## Timeframe consistency

The HCWs’ assignments are expected when graduating classes finish. **It is important to maintain a timeframe consistency within the tool**. For instance, if the user just wants to assign the current class graduates, (s)he should only include the relevant class in the input data . Also, the HCWs’ demand in the input data should only consider the demand for the current class (only the graduates that the user needs to assign at the moment).

## 

## Clearing input and results

In the configure section of the system, there is a “clear all” option that will clear all dynamic information in the system except the worker types, study locations and demand location names.

A “delete” functionality is available on most lists. It will allow users to delete unnecessary information in the system. **Once an item is deleted, it *cannot* be recovered.**

If input data elements are deleted, these deletions may distort the reports that would have been generated from running the allocation tool, as these data elements may have been referenced in other various reports.

## System Configuration

Next, we describe the content of each of the input screens:

1. General Settings
2. Worker Types
3. Worker Levels
4. Default Salaries
5. Demand Locations
6. Study Locations
7. Distance Lookup Table
8. Workers Settings

### 

### General Settings

**Input settings:** settings regarding the data input that apply generally to all other input data (unless otherwise specified). In the database, this table is named “setting.”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Num. | Title | User Input/ Tool Formula | Description | Data Validation |
| 1 | Haversine formula | Input | The default distances computed in this tool are based on the Haversine distance between the coordinates of two locations. It is an "as the crow flies" distance, ignoring hills, road networks and conditions, etc. This is a default correction of this distance, e.g., a factor greater than 1 for a longer “real road” distance. That is, Real distance approximation equals to the Haversine distance approximation times this factor. | Greater than or equal to 1. |
| 2 | Total available budget | Input | Budget to cover salaries for all the locations. If there is not a budget, leave it blank. | Blank or an integer greater than or equal to 0. |
| 3 | Maximum weight allowed for workers' preferences | Input | Maximum weight or preference score that can be assigned to a HCW’s location preference. | An integer greater than or equal to 0. |
| 4 | Default Penalty Unfulfilled Demand | Input | Default penalty for all demand locations can be set here. | An integer greater than or equal to 0. |
| 5 | Tool Currency | Input | Name of Currency used for budgets and salaries |  |
| 6 | Number of Preferences allowed | Input | Number of preferences allowed for each graduate can be assigned here | An integer greater than or equal to 1. |
| 7 | Platform | Input | Specify the operating system of the server | Either Linux or Windows. |

**Optimization model settings:** settings regarding how long the model is allowed to run in GLPK.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Num. | Title | User Input/ Tool Formula | Description | Data Validation |
| 1 | Maximum running time (minutes) | Input | When to stop running the model in GLPK (it is possible that the solver has not found the optimal solution yet). Recommended at least 60 minutes. | An integer greater than or equal to 1. |
| 2 | Optimality gap | Input | The model solution should be at least this close to the optimal (best) solution in order to stop running the model, before the maximum running time is achieved. The larger this percentage, the faster the model finishes; however, the solution might be further from the optimal. Recommended range: 1% to 10%. | From 0% to 100% |

### Worker Types - In the database, this table is named “worker\_type.”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Num. | Title | User Input/ Tool Formula | Description | Data Validation |
| 1 | Num. | Formula | It is just a consecutive number for the worker types. | No applicable. |
| 2 | Worker Type | Input | The list of different **demanded** worker types or categories (e.g., General Nurse, Medical Doctor, etc.). | No restrictions. |

### Worker Levels - In the database, this table is named “worker\_level.”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Num. | Title | User Input/ Tool Formula | Description | Data Validation |
| 1 | Num. | Formula | It is just a consecutive number for the worker levels. | No applicable. |
| 2 | Worker Salary Level | Input | Corresponds to the HCW salary group. Indicates the career and salary level. It is used to compute default salaries. For the recent graduates, it is expected to only include entry-level HCWs, so it may not be needed to specify the salary level and include just the career. E.g.,*Senior, Junior,* etc. | No restrictions. |
| 3 | Worker Level Description |  | Description of the worker level |  |

### Worker Salaries- In the database, this table is named “worker\_salary.”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Num. | Title | User Input/ Tool Formula | Description | Data Validation |
| 1 | Num. | Formula | It is just a consecutive number for the worker type and worker level combinations for default salaries. | No applicable. |
| 2 | Worker Salary Level | Input | A worker level (for salary purposes). | It should be selected from the list defined in “Worker Levels.” |
| 3 | Worker Type | Input | Aworker type or category. It needs to be specified only if the salary is different for a specific worker level and worker type combination. **If the salary for a worker level is a default for all the workers, independently of the workers' type, leave it blank.** | It should be selected from the list defined in “Worker Types” or leave blank if the salary is default for all the worker types. |
| 5 | Default Salary | Input | The default (average) salary for this combination of worker type and level. **If the worker type is blank, then it is the default for all the worker types**. For the recent graduates, it generally corresponds to the entry-level salary for the given career. | An integer greater than or equal to 0. |

### 

### Demand Locations - In the database, this table is named “demand\_location.”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Num. | Title | User Input/ Tool Formula | Description | Data Validation |
| 1 | Num. | Formula | It is just a consecutive number for the demand locations. | No applicable. |
| 2 | Demand Location | Input | The list of locations where the HCWs can be assigned. E.g., Arusha, Kigoma, etc. | No duplicate demand location name. |
| 3 | Coord. N | Input | Enter location's latitude in degrees. Required only to compute default distances. | Between -90 and 90. |
| 4 | Coord. W | Input | Enter location's longitude in degrees. Required only to compute default distances. | Between -180 and 180. |
| 5 | Budget | Input | Budget to cover salaries for the HCW assignments to the demand location. If there is not a budget, leave blank. | Blank or nn integer greater than or equal to 0. |
| 6 | Penalty for Unfulfilled Demand | Input | The penalty for not fulfilling the location's demand. It is applied to the percentage or fraction of unfulfilled demand. **The larger the penalty, the more important it is to fulfill the location’s demand.** Note that this balances with the preference scores/weights of the workers' preferences. | An integer greater than or equal to 0. |
| ≥7 | Demand by Worker Type | Input | Enter demand in number of workers, for each location, for each worker type. The worker types appear in the same order that they were entered in “Worker Types.” Below the worker type title, the user can see the total demand for all the locations computed. Include only the demand that needs to be fulfilled at the time—for instance, the demand to be fulfilled given the workers that graduate in a given semester. See the “Timeframe consistency” section. | An integer greater than or equal to 0. |

### Study Locations - In the database, this table is named “location.”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Num. | Name | User Input/ Tool Formula | Description | Data Validation |
| 1 | Num. | Formula | It is just a consecutive number for the study locations. | No applicable. |
| 2 | Study Location | Input | List of HCWs study locations. If these are the same than those in the list in "Demand Locations," just click on the button above to copy the information. | No duplicates Study locations. |
| 3 | Coord. N | Input | Enter location's latitude in degrees. Required only to compute default distances. | Between -90 and 90. |
| 4 | Coord. W | Input | Enter location's longitude in degrees. Required only to compute default distances. | Between -180 and 180. |

### Distances Lookup Table - In the database, this table is named “distance.”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Num. | Title | User Input/ Tool Formula | Description | Data Validation |
| 1 | Num. | Formula | It is just a consecutive number for the study and demand locations pairs. | No applicable. |
| 2 | Study Location | Input | Enter a worker study location or origin. It is only required if exact distances between study and demand locations can be given. | It should be selected from the list defined in “Study Locations”. |
| 3 | Demand Location | Input | Enter a demand location or destination. It is only required if exact distances between study and demand locations can be given. | It should be selected from the list defined in “Demand Locations”. |
| 5 | Road Distance (km) | Input | Enter the road distance between the two locations. Enter 0 if they are the same locations. | An integer greater than or equal to 0. |

### Worker Settings - In the database, this table is named “graduate.”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Num. | Title | User Input/ Tool Formula | Description | Data Validation |
| 1 | Num. | Formula | It is just a consecutive number for the HCW. | No applicable. |
| 2 | Graduate # | Input | It is a HCW unique identification number. If a unique identifier is not available, any **unique (positive integer)** number can be used, e.g., 1, 2, 3, …, 1000, 1001 (no need to be in order). Include only the workers to be assigned at the time (for instance, those graduating in the semester of interest). See the “Timeframe consistency” section. | A unique integer greater than or equal to 0. |
| 3 | Name | Input | HCW’s name. It is not necessary for the model or the tool, but should be included for reference. | No restrictions. |
| 4 | Gender | Input | Select “F” for “Female” and “M” for “Male.” Statistics in “Assignments by Location” can be given by gender. Also, the default preference scores can be given by gender. | Select from the list “F” or “M.” |
| 5 | Study Location | Input | Select the study location of the HCW. It is required as important HCW data, but it is not used in the model. It only used to calculate distances between study and demand locations’ assignments. | It should be selected from the list defined in “Study Locations.” |
| 6 | Coord. N | Formula | Shows the location's latitude in degrees, as entered in the “Study Location” worksheet. Required only to compute default distances. | Not applicable. |
| 7 | Coord. W | Formula | Shows the location's longitude in degrees, as entered in the “Study Location” worksheet. Required only to compute default distances. | Not applicable. |
| 8 | Worker Type | Input | Enter the HCW’s type or category **Do not leave this data field blank for any worker, or there will be an error.** | It should be selected from the list defined in “Worker Types.” |
| 9 | Worker Salary Level | Input | Corresponds to the HCW’s career and/or the salary group, and indicates the salary level. For the recent graduates, it is expected to only include entry levels. | It should be selected from the list defined in “Worker Levels.” |
| 10 | Default Salary | Formula | The default (average) salary for the combination of worker type and worker level. **If the salary is not available for this level and type combination, the formula looks for a default for the given worker level (with no specified worker type). If no default is found, it will show an “NA”. The salary for the corresponding worker level should be added to the worksheet “Default Salaries” or an adjusted salary for the HCW entered** **(otherwise it will be assumed it is 0).** The salary information is more relevant if there are any budget constraints. | Not applicable. |
| 11 | Adjusted Salary | Input | If the HCW salary is different from the default, the user should enter the actual salary here; otherwise, just leave blank. The salary information is more relevant if there are any budget constraints. | An integer greater than or equal to 0. |
| 13 | Potential Fixed Location | Input | Enter the worker’s potential fixed assignment location (e.g., their study location). | It should be selected from the list in "Demand Locations." |
| 14 | Assign to Fixed Location? | Input | Select “1” only if the model should force the worker’s assignment. **Make sure the location budget and demand are not exceeded with this fixed assignment (otherwise the optimization model might turn infeasible with no results). Check the worksheet "Fixed Assignments" to see if any budget or demand constraints are violated.**  “0” is the default if nothing is entered. | 1 or 0 (leaving it blank is equivalent to 0). |
| 15 | Do NOT Assign Outside of Preferences? | Input | Select “1” if the model should not assign the worker to a demand location outside his/her preferences. “0” is the default if nothing is selected. Note that not assigning the HCW to any location is still an option. | 1 or 0 (leaving it blank is equivalent to 0). |
| 16 | Number of Preferences | Formula | This column shows the number of preferences (1 to 10) that have been assigned a weight for the HCW. | Not applicable. |
| ≥17 | Preference # and Weight # | Input | We recommend including three preferences per worker, but the user can enter more or less. For each preference, the user must select the location of preference and the weight or preference score in the following column. The weights that are not completed can be filled out automatically using the macro button “Default Weights,” based on the data in the worksheet “General Settings.” If one location is selected more than once, we take into account the highest ranked selection.  **If there are no location preferences for the worker (or for all the workers), just leave these cells blank.** | If there are no preferences, leave blank. Otherwise, the preferred location(s) should be selected from the list in “Demand Locations.” The preference weight should be greater than or equal to 0. If the weight is blank, it is equivalent to 0. Also, the weight of any “Preference 1” should not exceed the maximum weight entered in “General Settings.” The weights of any preference (default or for a given HCW), should not exceed the weight of a previous preference (higher ranked). E.g., a “Preference 2” should not exceed the weight of a “Preference 1.” |
| 18 |  | Input | Date of Birth | Format dd/MM/yyyy |

### Fixed Assignments

For the outputs, we summarize the HCWs’ fixed assignments by location and worker type, to make sure they do not violate budget or demand constraints.

|  |  |  |
| --- | --- | --- |
| Num. | Title | Description |
| 1 | Num. | It is just a consecutive number for the demand location. |
| 2 | Demand Location | List of demand locations. Shown in the same order as in “Demand Locations.” |
| 3 | Allocated Budget by FIXED | Shows the total in salaries that were assigned to each location (and overall) by fixing HCWs to a particular location. |
| 4 | % Allocated Budget by FIXED Location | Shows the percentage of the available budget that was assigned to each location (and overall) by fixing some HCWs to a particular location. **If this percentage is greater than 100%,** **a warning (in red) will appear. The model will be infeasible, and a warning message will appear if the user tries to run the model, unless some of the workers are un-fixed (in “Worker Settings”).** |
| ≥5 | Fixed Assignments by Location and Worker Types | Shows the percentage of the demand for workers of a certain type that were assigned to each location by fixing some HCWs to a particular location**. If this percentage is greater than 100%, the model will be infeasible and a warning message will appear if the user tries to run the model, unless some of the workers are un-fixed (in “Worker Settings”).** |

### 

### Workers Assignments

Worker Assignments output shows the results of the assignments for all HCWs. We also include important data for each of the HCWs.

|  |  |  |
| --- | --- | --- |
| Num. | Title | Description |
| 1 | Num. | It is just a consecutive number of HCWs. |
| 2 | Unique identifier(UID) | HCW unique identification number. They appear in the same order as in “Worker Settings.” |
| 3 | Name | HCW name (from “Worker Settings”). |
| 4 | Location Assignment | Demand location assigned to the HCW. It is read from the GLPK output file. If there is no location, the HCW was not assigned. |
| 5 | Preference | Preference number (i.e., its rank) for the location assigned to the HCW. It is left blank if the location is not listed in the worker’s preferences or there is no location assigned. |
| 6 | Preference Assignment | Preference weight (i.e., score) for the location assigned to the HCW. It is left blank if the location is not listed in the worker’s preferences or there is no location assigned. |
| 7 | Gender | For those HCWs that were assigned (it will be blank otherwise), it’s the HCW’s gender (from “Worker Settings”). |
| 8 | Adjusted Distance (km) | For those HCWs that were assigned (it will be blank otherwise), the approximate distance between the HCW’s study and assigned location**. If the pair is found at “Distances Lookup Table,” the distance is reported from the table. Otherwise, a default distance is computed based on the Haversine formula, using the locations’ coordinates, with time as the adjusting factor in “General Settings”.** |
| 9 | Worker Type | For those HCWs that were assigned (it will be blank otherwise), HCW’s type (from “Worker Settings”). |
| 10 | Worker Level | For those HCWs that were assigned (it will be blank otherwise), HCW’s level (from “Worker Settings”). |
| 11 | Salary (Allocated) | For those HCWs that were assigned (it will be blank otherwise), HCW’s salary (from “Worker Settings”). The default salary is reported unless an adjusted salary is given for the worker. |

### 

### Demand Met by Location

The output of Demand met by location shows the results for the demand fulfillment for each location and worker type.

|  |  |  |
| --- | --- | --- |
| Num. | Title | Description |
| 1 | Num. | It is just a consecutive number for the demand location. |
| 2 | Demand Location | List of demand locations. Shown in the same order as in “Demand Locations”. |
| 3 | Worker Type | Worker Types saved in the database |
| 4 | Requested | Requested Number Of Workers |
| 5 | Available | Available Number of Workers to be allocated |

### 

### Assignments by Location

The output of Assignments by location show the statistics of the assignment results for each demand location (and overall for all locations at the top).

|  |  |  |
| --- | --- | --- |
| Num. | Title | Description |
| 1 | Demand Location | List of demand locations. Shown in the same order as in “Demand Locations”. |
| 2 | Workers Assigned | Number of HCWs assigned to the demand location. |
| 3 | Workers Requested | Number of HCWs requested by each demand location. |
| 4 | % Assigned vs. Total Num. of Workers | Percentage of HCWs assigned to the demand location with respect to the total number of HCWs assigned to the location. |
| 5 | Workers Assigned to Their Top Choice | Number of HCWs assigned to the demand location, such that the location was their first preference. |
| 6 | % Workers Assigned to Their Top Choice | Percentage of HCWs assigned to the demand location, such that the location was their first preference, with respect to the total number of HCWs assigned to the location. |
| 7 | Workers Assigned to Their Top 3 Choices | Number of HCWs assigned to the demand location, such that the location was within their top three preferred locations. |
| 8 | % Workers Assigned to Their Top 3 Choices | Percentage of HCWs assigned to the demand location, such that the location was their top three preferred locations, with respect to the total number of HCWs assigned to the location. |
| 9 | Allocated Budget | Total budget allocated as salaries to each location. |
| 10 | % Allocated of Total Budget | Percentage of the location’s budget allocated as salaries to each location. |

### 

### Assignments by Worker Types

The output of Assignments by worker types shows the statistics of the assignment results for each worker type (and overall for all worker types at the top). This worksheet can help to compare the total demand for each worker type, with the available number of workers of that type. They can help the user to adjust demand data if necessary.

|  |  |  |
| --- | --- | --- |
| Num. | Title | Description |
| 1 | Worker Type | List of worker types. Shown in the same order than in “Worker Types”. |
| 2 | Total Demand | Number of workers demanded for each worker type. |
| 3 | Total Number of Workers | Number of HCWs available for assignment for each worker type. |
| 4 | Workers Deficit | The number of additional HCWs required to fulfill all the demand. If there are more HCWs than the workers demand, it equals zero. |
| 5 | Workers Surplus | The number of available HCWs in excess of the demand. If there are more workers demanded than available, it equals zero. These HCWs cannot be assigned unless the demand is increased. |
| 6 | Workers Assigned | Number of HCWs assigned for each worker type. |
| 7 | Workers NO Assigned | Number of HCWs not assigned for each worker type. |

### 

### Preferences by Location

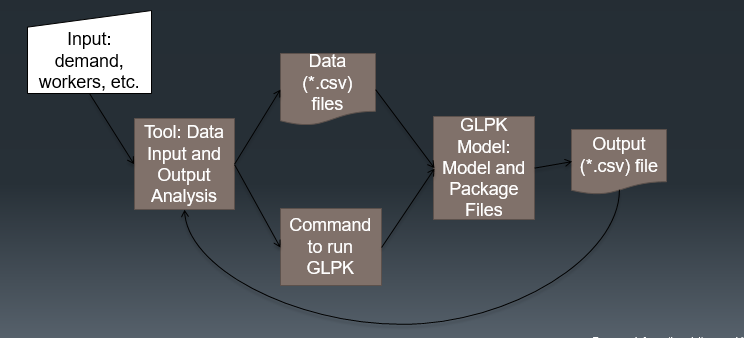
The output of preferences by location shows how popular the demand locations are in terms of how many times the HCWs selected them as a top or top-three choice.

|  |  |  |
| --- | --- | --- |
| Num. | Title | Description |
| 1 | Demand Location | List of demand locations. Shown in the same order as in “Demand Locations”. |
| 2 | Selected in Preferences | The number of times the demand location was selected in the HCWs preferences. |
| 3 | % Preference Selections | The percentage of times when the demand location was selected in the HCWs’ preferences, with respect to the total number of selections. |
| 4 | Selected as Top Choice | The number of times the demand location was selected as a top choice in the HCWs’ preferences. |
| 5 | % Selections as Top Choice | The percentage of times when the demand location was selected as a top choice in the HCWs preferences, with respect to the total number of top choice selections. |
| 6 | Selected as Top 3 Choice | The number of times the demand location was selected as a top-three choice in the HCWs preferences. |
| 7 | % Selections as Top 3 Choice | The percentage of times when the demand location was selected as a top-three choice in the HCWs’ preferences, with respect of the total number of top-three choice selections. |

# 

# Running the model

A general diagram of how to prepare and run the workforce model is given below:



First, the user enters the input data into the tool. Once the data input is ready, the user should run the optimization model by clicking “Run GLPK.” A message will appear showing that the Workforce Allocation Optimization Tool is running.

1. After clicking **“**Run GLPK,**”** the Workforce Allocation Optimization Tool will generate the (input) data files required by the GLPK. The model will use the input file, the model implementation (“Model.txt”), and the GLPK package files (“glpsol.exe” and “glpk\_4\_57.dll”) to do the actual allocation. **When the model has finished, a message read from the** “log.txt” file will be shown.
   1. If an optimal integer solution was found, a message appears: "Allocation Complete - OPTIMAL SOLUTION FOUND.” After the model is run successfully, GLPK produces an output file with the model results “results\_x.csv,” and this file’s results are read and saved in the database. The Workforce Allocation Optimization Tool will then display the results as a set of different web results.
   2. If the time limit was exceeded and no *optimal* solution was found, a message appears: “Allocation Complete – More Time Needed.” Increase the maximum running time or the optimality gap (or both) in the worksheet “General Settings.”
   3. Finally, if there was another type of issue, and no optimal solution was found, a message appears: "Allocation Failed, Errors Found.” Check the GLPK log file for more information. Check the section of potential errors in this documentation. Clicking the error on the system will display the results of the log.

## 

## Data files

The data files are .csv files generated using the tool by clicking on the button “Run GLPK.” They contain data based on the tool input that are read by GLPK to build the optimization model. **Each data file represents a data table, with the first row containing the field names. These field names should not be changed, nor should the data structure of each file. The following file names also should not be changed:**

1. “demand.csv”
2. “general.csv”
3. “locations.csv”
4. “preferences.csv”
5. “types.csv”
6. “workers.csv”

### demand.csv

This file includes the demand for each location and worker type. **There should be a row for each location (as listed in “locations.csv”) and worker type (as listed in “types.csv”) combination.**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Field | Description | Data Type |
| 1 | Location | **Demand Location.** | Characters |
| 2 | Type | **Worker type or category.** | Characters |
| 3 | Demand | Number of demanded HCW. | Numbers (integer) |

### general.csv

This file includes overall settings or parameters for the optimization. **There is only one setting in this file, “totalbudget,” which represents the total budget available** (considering all the demand locations). **If there is no such budget, the file includes the legend “NA” instead of the value. The reason is that we cannot leave fields blank, since we would get an error in GLPK.**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Field | Description | Data Type |
| 1 | Setting | Name of the setting. | Characters |
| 2 | Value | Value of the setting. | Numbers (integer) or “NA” |

### locations.csv

This file includes the list of locations, their budgets and their penalties in case of unfulfilled demand. There should be a row for each demand location. If there is not a budget for the location, the file includes the legend “NA” instead of the value. The reason is that we cannot leave fields ‘blank’, since the user would get an error from GLPK.

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Field | Description | Data Type |
| 1 | Location | Demand Location. | Characters |
| 2 | Budget | Worker type or category. | Numbers (integer) or “NA” |
| 3 | Penalty | Number of demanded workers. Note that locations with higher priority should have higher weights. | Numbers (integer) |

### preferences.csv

This file includes the list of location preferences for each HCW and its corresponding weight or preference score. There should a row for each time a different location (as listed in “locations.csv”) is reported as a preference by the HCW. For instance, if a worker reports three different preferences, there would be three rows for that worker. If the worker repeats locations in his/her preferences, the worker/location pair is only included once, with the higher weight. The reason is that if we repeat a worker/location pair, GLPK will not be able to know which the correct weight is, and the user would get an error. If there is not a weight particular to the HCW, a default weight should be considered (according to what is specified in the worksheet “General Settings”), so that there are not blanks in the weights values. If a HCW does not have any preferences, then this worker should not be listed in this file. If none of the HCWs have a preference, then this file still needs to be included, but it will only list the fields’ titles and no additional data.

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Field | Description | Data Type |
| 1 | Worker | HCW’s unique identifier. | Numbers (integer) |
| 2 | Location | A preferred demand location for the HCW. | Characters |
| 3 | Weight | The associated weight or preference score for the assignment of the HCW to the demand location. Note that higher preferences should have higher weights. | Numbers (integer) |

### Types.csv

This file includes the list of worker types or occupations/categories. **There should be a row for each worker type.**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Field | Description | Data Type |
| 1 | Type | Worker type. | Characters |

### Workers.csv

This file includes the list of all HCWs, their type, salary and particular assignment constraints, such as which potential location the HCW may be fixed to, if the HCW should be fixed to that location, or if the HCW should only be assigned to one of his/her preferred locations or not be assigned at all. **There is a row for each HCW.** **If there is not a potential fixed location for the HCW, the file includes the legend “NA” instead of the value. The reason is that we cannot leave fields ‘blank’, since the user would get an error from GLPK.**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Field | Description | Data Type |
| 1 | Worker | HCW’s unique identifier. | Numbers (integer) |
| 2 | Type | HCW’s type or category. | Characters |
| 3 | Salary | HCW’s salary. | Numbers (integer) |
| 4 | Fixed | Equals to 1 if the HCW is fixed to a location and 0 otherwise. | 0 or 1 |
| 5 | FixedLocation | The potential fixed demand location. | Characters or “NA” |
| 6 | OnlyPreferences | Equals to 1 if the HCW can only be assigned to one of his/her preferences and 0 otherwise. | 0 or 1 |

## GLPK Model

**Running GLPK**

The model is run in GLPK by clicking on the button “Run GLPK.” This opens, runs and solves the optimization model declared in “Model.txt” using the .csv files for data input. Parts of these instructions include the minimum gap and the maximum running time.The macro also generates a file “RunGLPK.bat,” which includes the same shell run commands. The contents of this file are like the following:

“glpsol --mipgap 0.01 --tmlim 21600 -m Model.txt --log log.txt”

The executable’s name file is “glpsol.” Then, we indicate that the minimum gap is 1% or 1/100=0.01 “—mipgap 0.01”, and that the time limit is 21600 seconds (360 minutes) “—tmlim 21600”. **The time limit should be in seconds, and the gap should be a decimal number between 0 and 1 (recommended range is between 0.01 and 0.1). Note that GLPK assumes that the user uses a period to indicate decimals, and not a comma. If the percentage is fractional, the tool rounds it to the closer whole number percentage (e.g., 1.5% to 2%, or 2.2% to 2%) when creating these shell commands**. Then, we indicate that the model is found in “Model.txt” with “-m Model.txt”. Finally, we instruct to create a log file name “log.txt” with “—log log.txt”.

The tool automatically creates a file “RunModel.bat” with these shell commands after clicking on the button “Run GLPK.”

## 

**GLPK Output**

**After the minimum gap is achieved or the model has been running longer than the time limit (whatever happens first), GLPK stops and displays the best solution it found.** At the end of “Model.txt,” we also instruct GLPK to create a file with the details of the HCWs’ assignments. This output file is named “results\_x.csv.” This output is then read back into the database automatically.

This output file includes the list of HCWs assignments to the demand locations. **There is a row for each assignment. Only those HCWs that were assigned are included here (those HCWs that were not assigned do not appear here).** **Reasons for not assigning all HCWs include: there is surplus of workers and demand needs to be increased, there are too many restrictions on the assignment of workers (e.g., workers that can only be assigned to one of his/her preferences), or the penalty for unfulfilled demand is too low (see worksheet “Assignments by Worker Types” documentation for more details).**

**If the user wants to alter the solution given by GLPK, he/she can do it in this file. However, if the user alters this file, (s)he must remember to only list each HCW at most once.**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Field | Description | Data Type |
| 1 | Worker | HCW’s unique identifier. | Numbers |
| 2 | Location | Demand location assignment for the HCW. | Characters |

**Potential errors when running the model with GLPK**

Some of the common errors are listed below. **The user can check for errors detected by GLPK in the “log.txt” file**. Some of these errors will stop the model’s running process. There are several validations in the tool to avoid most of these errors.

* The input is incomplete. For instance, information like worker’s type is not given for a HCW.
* The names for workers, worker types, demand locations, etc., are not consistent across the data files.
* The user made some fixed assignments that are not feasible. For instance, budgets are violated or these fixed assignments surpass the demand of the location for a given worker type.
* Some of the data files, GLPK package files or the “Model.txt” are missing or are not included in the same directory.
* Need to close data \*.csv files that were opened by the user.
* The user by mistake changed the name of a file.
* The user by mistake changed the title of a column of the input data files.
* The user used forbidden symbols in names e.g “ ’, @!@#$%^&\*()+-“.
* GLPK needs the numbers in the American standard, using a decimal point (not a decimal comma). In the tool, we try to ensure this by adding the instruction to use period as decimal separator, and commas as thousands separators.
* The user used negative values for demand weights, salaries, penalties or budgets.
* The user used values other than “0” or “1” for the “Fixed” and “OnlyPreferences” fields in “workers.csv.”
* There are some blank values in the data files. **Blank values are only allowed at the end of the data files or data tables; otherwise, GLPK will report an error reading the data**. For budgets or fixed locations, “NA” should be used in case there is not data. Also, zero can be used for other cases, but never a blank value.
* The user used other values than “NA” when there was no budget available in “general.csv” and “demand.csv”, or no potential fixed location in “workers.csv”.
* We have repeated fields or field combinations (that create sets in the model). For instance, a worker’s unique identifier and demand location combination is listed more than once in the “Preferences.csv” data file.

## Recommend Security Features

In summary, this section outlines the security features recommend by for hosting the Workforce Allocation Optimization Tool.

### Physical Access

The Workforce Allocation Optimization Tool server should be housed in a server room that is not easily accessible. Only authorized individuals should gain access through the use of a physical mechanism, including locks.

### Application/Software Level

**Firewall**

A firewall should be installed between the point of internet connection and server. The server's operating system should also have an inbuilt firewall. The firewall allows only a certain type of traffic/protocols through it and all the unnecessary ports are closed/prohibited or a default deny.

**Database Level**

The database root password is not used by any application. All connecting applications have their own respective database accounts with the relevant permissions or limits. There is no database server web Graphical User Interface (GUI) tools installed on the publicly accessible servers.

The following are some of the security features within the Workforce Allocation Optimization Tool:

**1. User passwords**

Currently, all users must have passwords in order to gain access to the system. Without a password, you cannot login to the system. Each user's password should be of a certain length (at least six mixed case characters) and this password is validated to ensure that it is not too simple. All passwords are obfuscated/hashed by means of high-grade SHA-512 one-way encryption with a variable length salt to prevent against dictionary attacks. This basically means that nobody can figure out a password of another user if somehow (s)he can gain access to the database.

**2. User sessions**

Every time a user logs in to the system, (s)he starts a session. Only during a session is a user able to use the functions available in the system. This session will automatically elapse if the user leaves his/her computer for more than five minutes, requiring the user to login again if (s)he intends to continue use of the system. This prevents someone accessing his/her computer in the absence of the logged in user.

**3. User roles**

Each user has a specific role. Users assigned to a specific role have specific permissions in the system assigned to them as well. For example, only the system administrator can assign/remove roles from users.

**User/Group Access Permissions/Logs**

Processes that run on the server are initiated and owned by defined system users, and all the processes are logged.

**Transport Layer / SSL /Encryption**

The Workforce Allocation Optimization Tool should make use secure sockets layer (SSL) for encryption where a system is required to be accessible from the internet.

## System Installation

The Allocation Tool uses the "LAMP" stack which is a group of open source software that is typically installed together to enable a server to host dynamic websites and web apps. The application is hosted on an Apache web server and uses a MySQL database and is developed in PHP. The Allocation Tool runs on both Linux and Windows.

### Installation on a Linux Ubuntu Server

**Install Apache**

sudo apt-get update

sudo apt-get install apache2

**Open Firewall**

sudo ufw app info "Apache Full"

sudo ufw app list

**Install MySQL**

sudo apt-get install mysql-server

Create the allocation tool database in mysql and import into mysql

**Install PHP**

sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql

Restart Apache

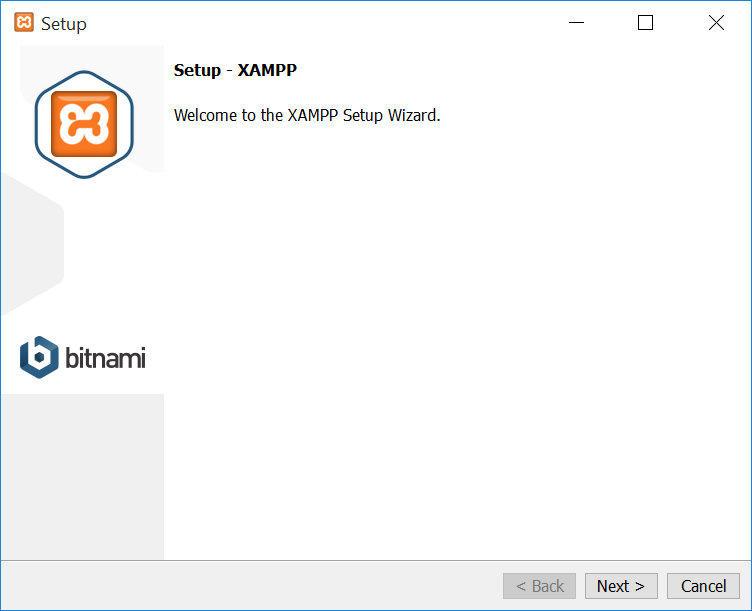
sudo systemctl restart apache2

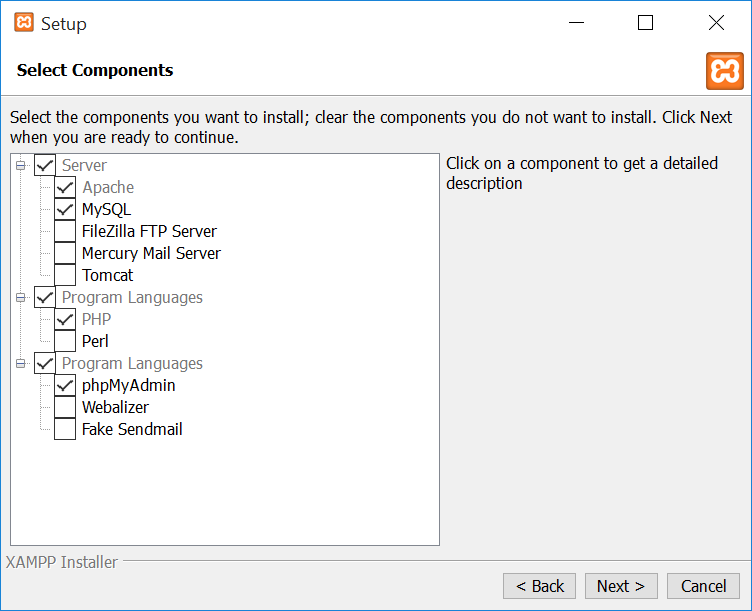
Copy the allocation tool folder into /var/www/html

Open in browser: localhost/allocationtool

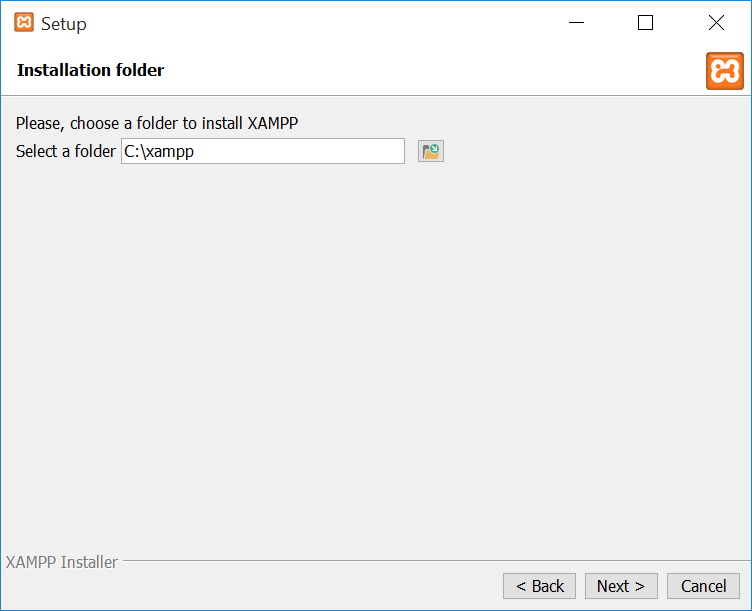
### Installation on a Windows Server

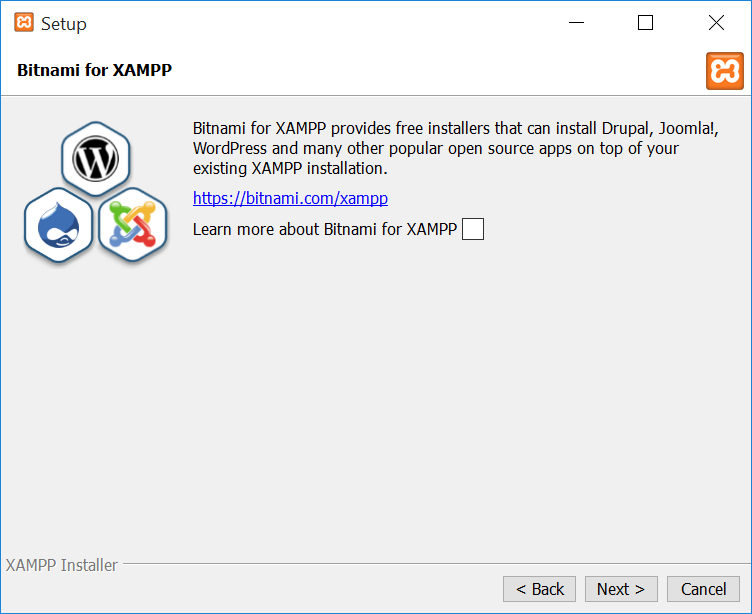
Install XAMPP, which is an easy to install Apache distribution containing MySQL, PHP, and Perl. Download XAMPP for the Apache Website. Once file has been download, double click the XAMPP installation file. This will start the XAMPP set up wizard. Click Next to start the process, follow the instructions to complete the process

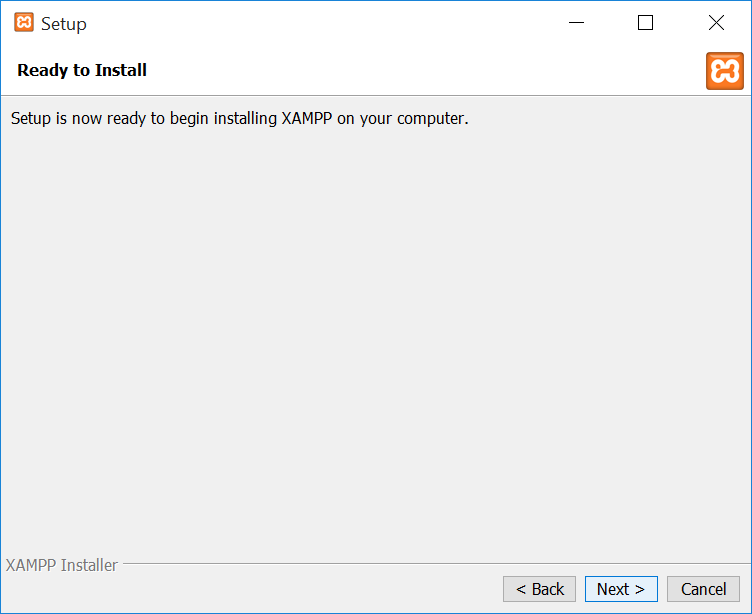


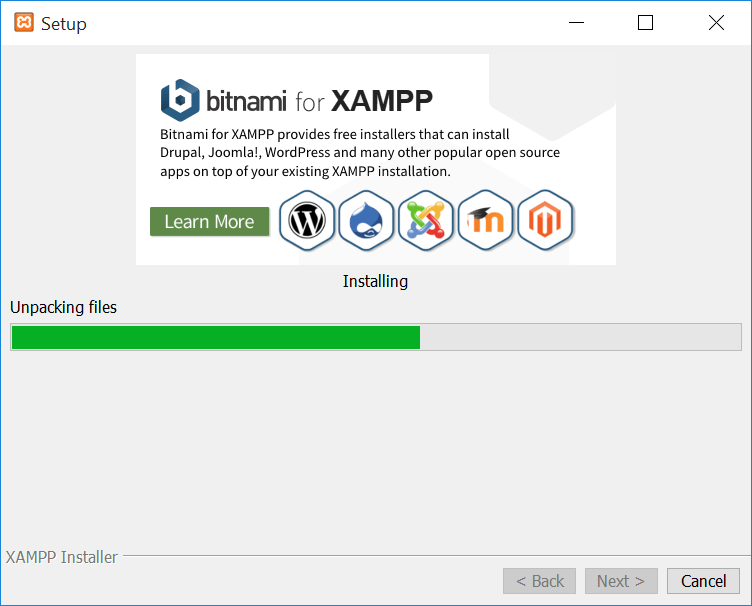


Select the components you would like to install. The tool requires MySql and Apache. Click next and select the destination folder where XAMPP will be installed.

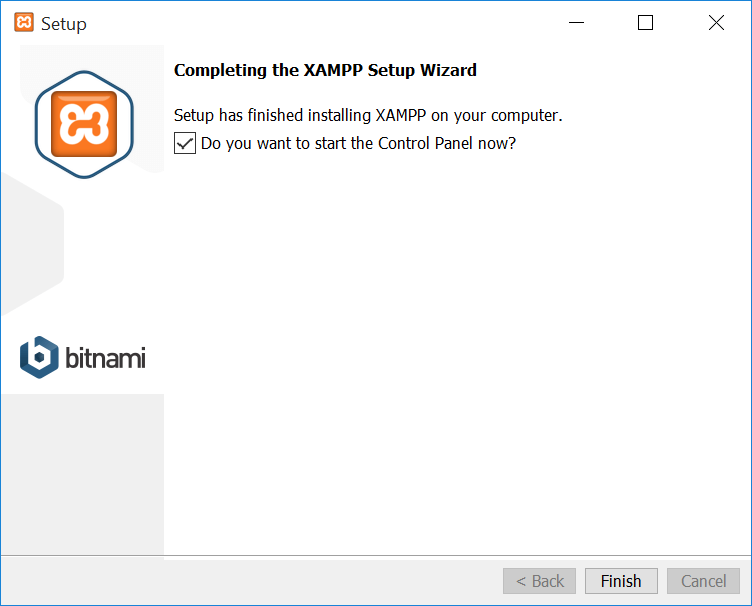




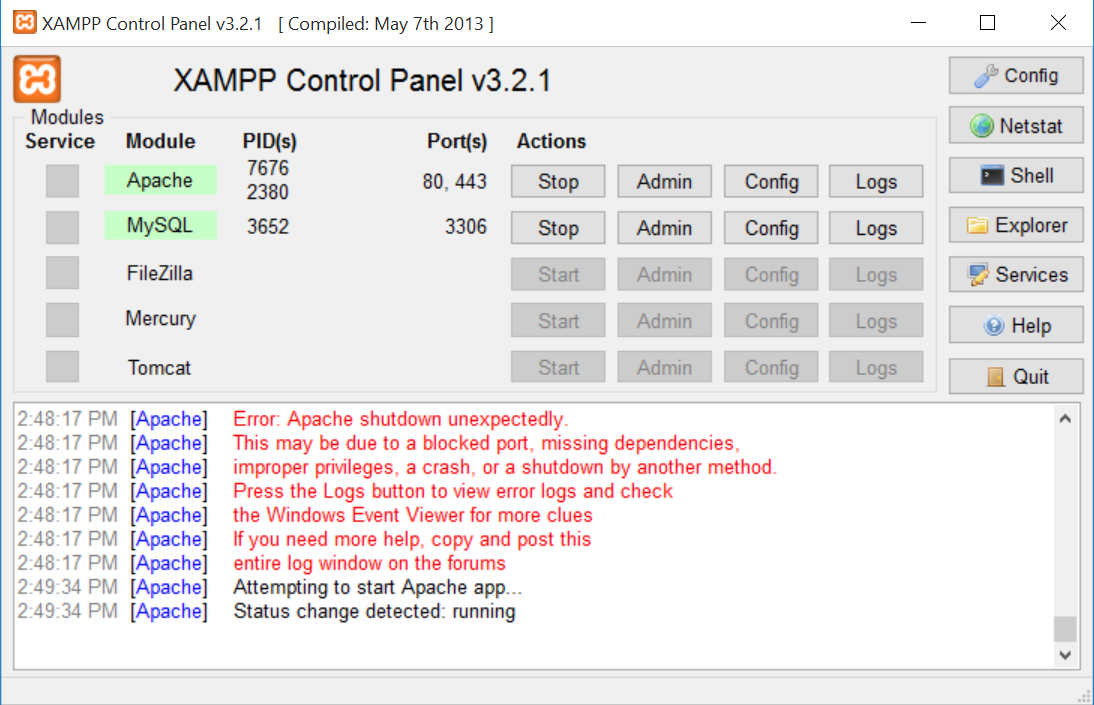




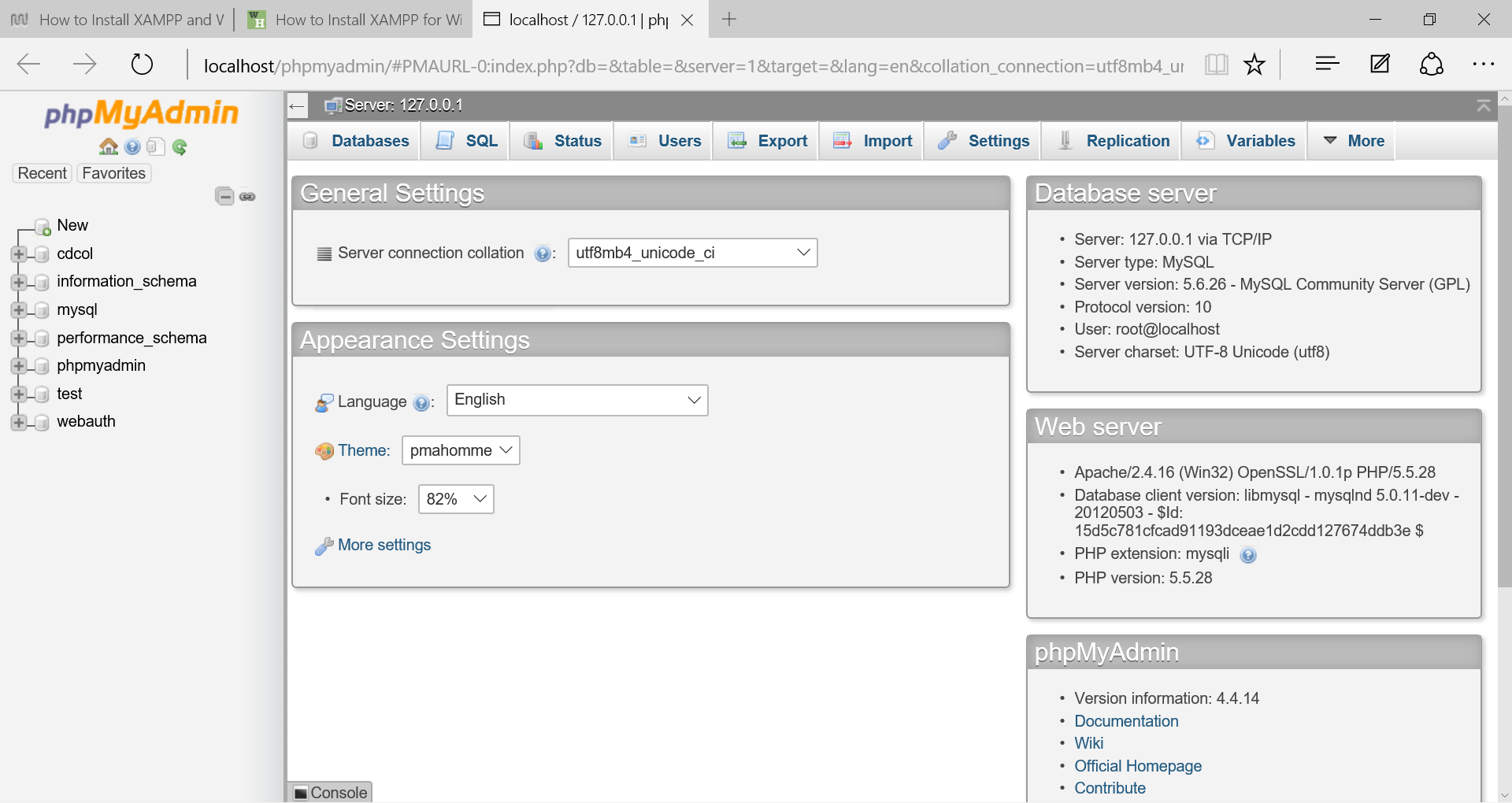
Once installation is complete, Click finish to open XAMMP Control panel



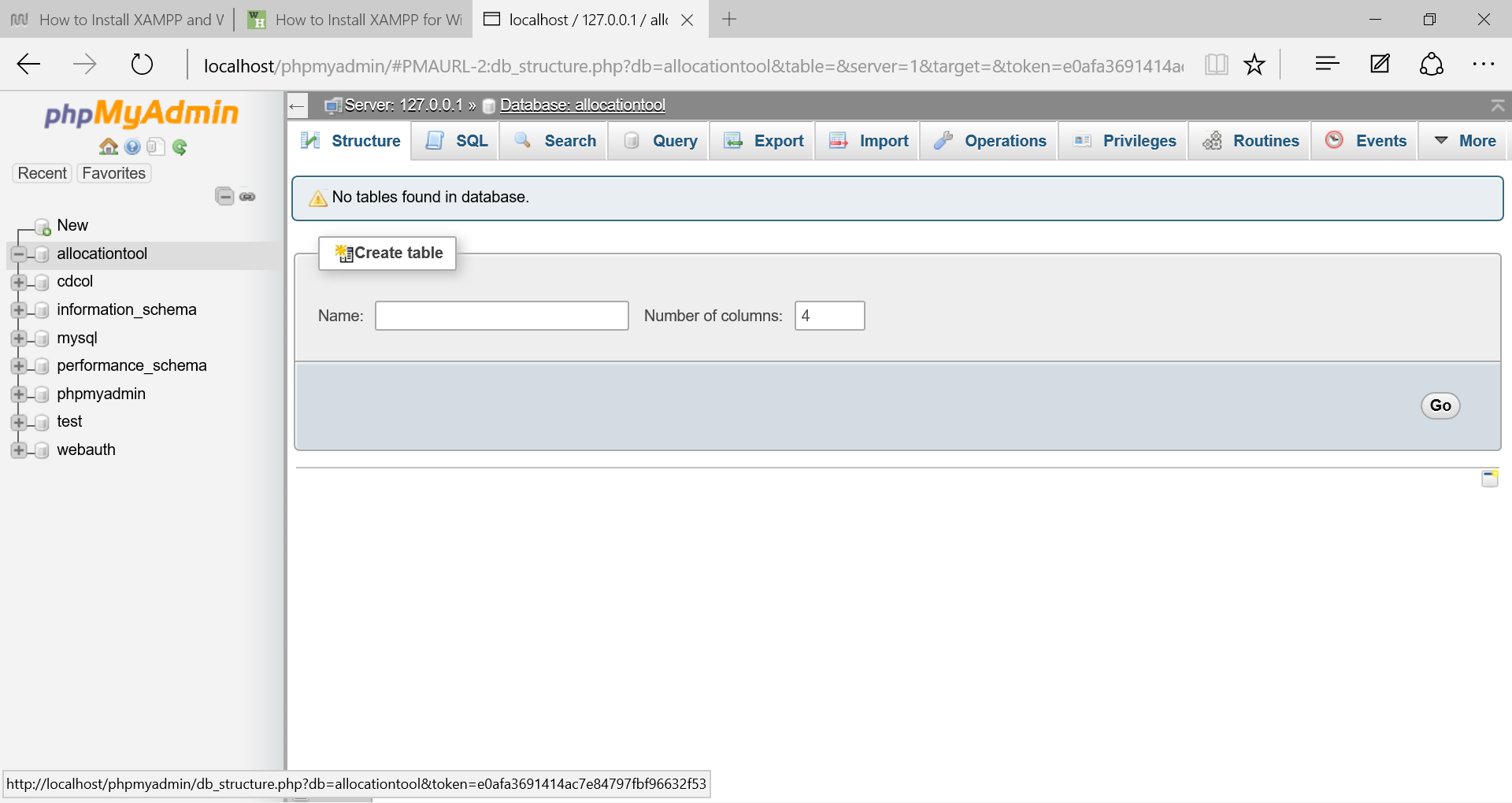
Start Apache and MySql



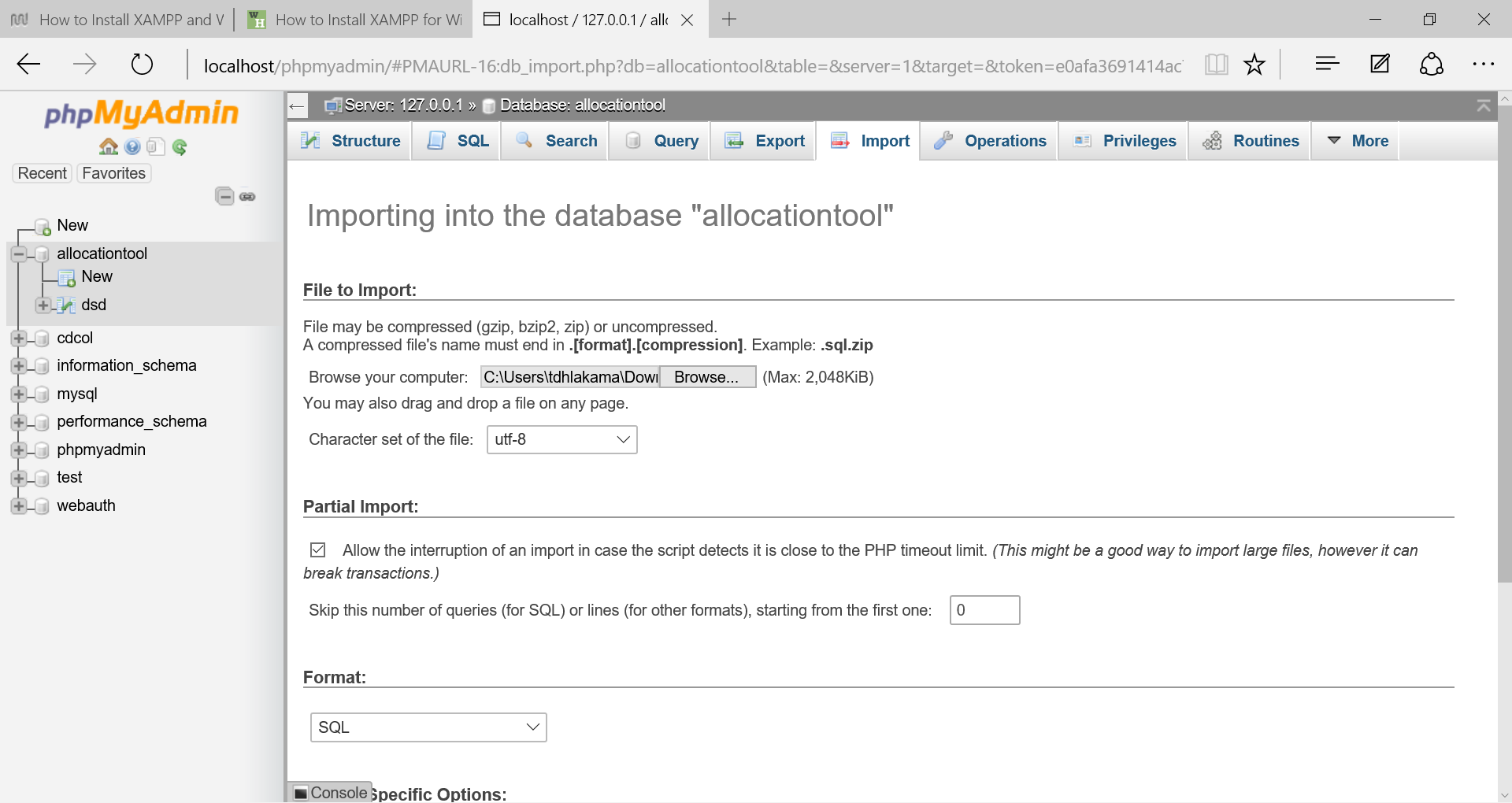
Click Admin on MySQL



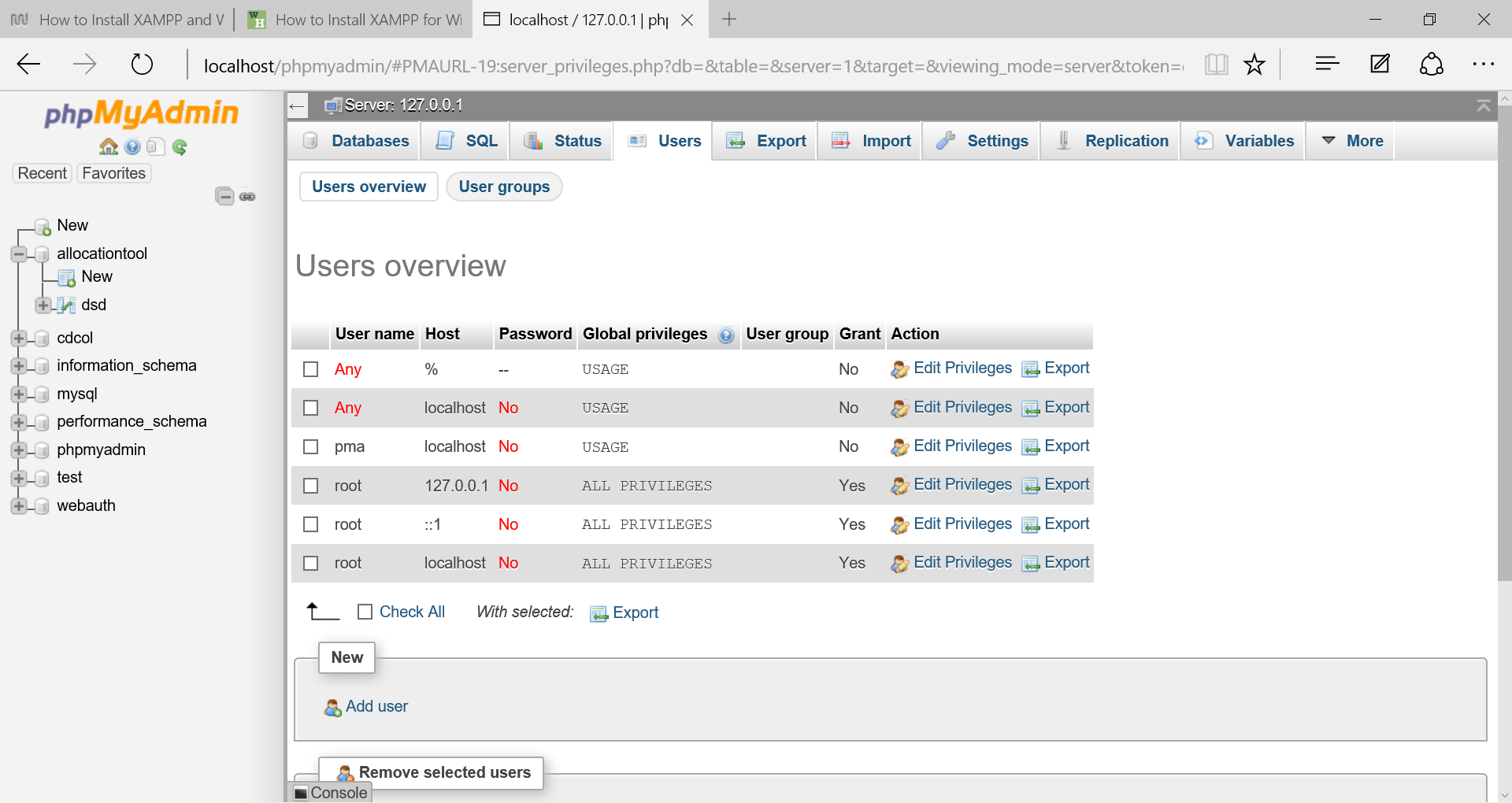
Under databases, create a new database called allocationtool



Go to import and import the allocationtool database

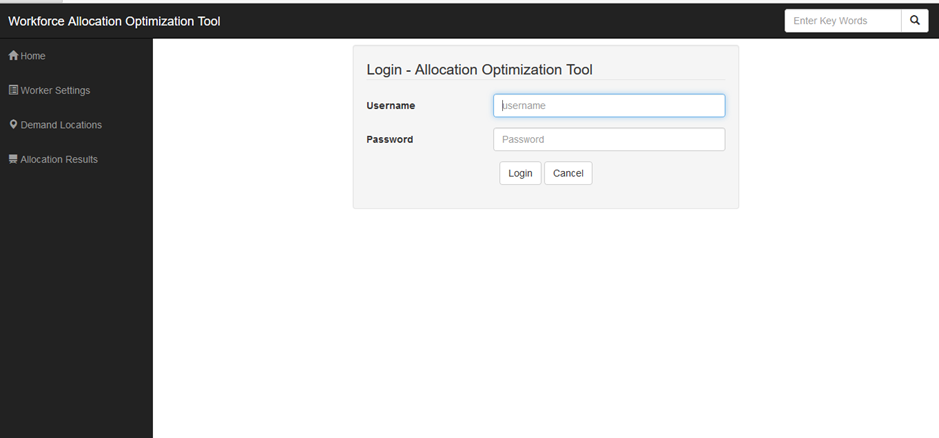


Under Users, create the application user name required by the allocationtool to access MySql.



Next Copy the allocationtool folder in to c: /htdocs/html/

Then open in browser and type localhost/allocationtool, The login page should appear.



## “ModelDB.txt”

##Declare parameters and set and read data form input files

#Read general data such a global budget constraint

set G;

param GeneralSetting{g in G}, symbolic;

table t IN

'ODBC' 'FILEDSN=.\workforce\_allocation.dsn'

'general' :

G<- [Setting], GeneralSetting ~ Value;

#Read the locations and locations data

set J;

table t IN

'ODBC' 'FILEDSN=.\workforce\_allocation.dsn'

'locations' :

J<- [Location];

param LocationBudget{j in J}, symbolic;

table t IN

'ODBC' 'FILEDSN=.\workforce\_allocation.dsn'

'locations' :

[Location], LocationBudget ~ Budget;

param LocationPenalty{j in J};

table t IN

'ODBC' 'FILEDSN=.\workforce\_allocation.dsn'

'locations' :

[Location], LocationPenalty ~ Penalty;

#Read the types of demand

set T;

table t IN

'ODBC' 'FILEDSN=.\workforce\_allocation.dsn'

'types' :

T <- [Type];

#Read the workers list and workers data

set I;

param Workers{i in I};

table t IN

'ODBC' 'FILEDSN=.\workforce\_allocation.dsn'

'workers' :

I<- [Worker];

param WorkerType{i in I}, symbolic;

table t IN

'ODBC' 'FILEDSN=.\workforce\_allocation.dsn'

'workers' :

[Worker], WorkerType ~ Type;

param WorkerSalary{i in I};

table t IN

'ODBC' 'FILEDSN=.\workforce\_allocation.dsn'

'workers' :

[Worker], WorkerSalary ~ Salary;

param WorkerFixedLoc{i in I}, symbolic;

table t IN

'ODBC' 'FILEDSN=.\workforce\_allocation.dsn'

'workers' :

[Worker], WorkerFixedLoc ~ FixedLocation;

param WorkerFixed{i in I};

table t IN

'ODBC' 'FILEDSN=.\workforce\_allocation.dsn'

'workers' :

[Worker], WorkerFixed ~ Fixed;

param WorkerOnlyPref{i in I};

table t IN

'ODBC' 'FILEDSN=.\workforce\_allocation.dsn'

'workers' :

[Worker], WorkerOnlyPref ~ OnlyPreferences;

#Read demand data

param Demand{j in J, t in T};

table t IN

'ODBC' 'FILEDSN=.\workforce\_allocation.dsn'

'demand' :

[Location,Type], Demand ~ Demand;

#Read the list of preferences for the workers

set Preference\_set, dimen 2;

param Weight{(i,j) in Preference\_set};

table t IN

'ODBC' 'FILEDSN=.\workforce\_allocation.dsn'

'preferences' :

Preference\_set <-[Worker,Location], Weight ~ Weight;

## Declare the decision variables. Only those variables that can be used are declared.

var x{i in I, j in J: (WorkerFixed[i]<>0 and WorkerFixedLoc[i]=j) or (WorkerFixed[i]=0 and WorkerOnlyPref[i]<>0 and (i,j) in Preference\_set) or (WorkerFixed[i]=0 and WorkerOnlyPref[i]=0)}, binary, >=0;

var a{j in J, t in T}, >=0;

## Declare the objective function

maximize Z: sum{(i,j) in Preference\_set: WorkerFixed[i]=0}(Weight[i,j]\*x[i,j]) - (sum{j in J, t in T}(LocationPenalty[j]\*(1-a[j,t])))

;

## Declare the model's constraints

s.t. fixedlocation{i in I, j in J: (WorkerFixed[i]<>0 and WorkerFixedLoc[i]=j)}: x[i,j]=1;

s.t. upperboundalpha{j in J, t in T}: a[j,t]<=1;

s.t. onelocation{i in I}: sum{j in J: (WorkerFixed[i]<>0 and WorkerFixedLoc[i]=j) or (WorkerFixed[i]=0 and WorkerOnlyPref[i]<>0 and (i,j) in Preference\_set) or (WorkerFixed[i]=0 and WorkerOnlyPref[i]=0)}x[i,j]<=1;

s.t. budgetlimit{j in J: LocationBudget[j] <>'NA'}: sum{i in I: (WorkerFixed[i]<>0 and WorkerFixedLoc[i]=j) or (WorkerFixed[i]=0 and WorkerOnlyPref[i]<>0 and (i,j) in Preference\_set) or (WorkerFixed[i]=0 and WorkerOnlyPref[i]=0)}x[i,j]\*WorkerSalary[i]<=LocationBudget[j];

s.t. totalbudgetlimit{g in G: g= 'totalbudget' and GeneralSetting['totalbudget'] <>'NA'}: sum{i in I, j in J: (WorkerFixed[i]<>0 and WorkerFixedLoc[i]=j) or (WorkerFixed[i]=0 and WorkerOnlyPref[i]<>0 and (i,j) in Preference\_set) or (WorkerFixed[i]=0 and WorkerOnlyPref[i]=0)}x[i,j]\*WorkerSalary[i]<=GeneralSetting['totalbudget'];

s.t. limitdemand{j in J, t in T}: sum{i in I: (WorkerFixed[i]<>0 and WorkerFixedLoc[i]=j and WorkerType[i]=t) or (WorkerFixed[i]=0 and WorkerOnlyPref[i]<>0 and (i,j) in Preference\_set and WorkerType[i]=t) or (WorkerFixed[i]=0 and WorkerOnlyPref[i]=0 and WorkerType[i]=t)}x[i,j]<=Demand[j,t];

s.t. meetdemand{j in J, t in T}: sum{i in I: (WorkerFixed[i]<>0 and WorkerFixedLoc[i]=j and WorkerType[i]=t) or (WorkerFixed[i]=0 and WorkerOnlyPref[i]<>0 and (i,j) in Preference\_set and WorkerType[i]=t) or (WorkerFixed[i]=0 and WorkerOnlyPref[i]=0 and WorkerType[i]=t)}x[i,j]>=a[j,t]\*Demand[j,t];

## Solve the model

solve;

#Create table with results of those workers assigned and their locations

table t { i in I, j in J: ((WorkerFixed[i]<>0 and WorkerFixedLoc[i]=j) or (WorkerFixed[i]=0 and WorkerOnlyPref[i]<>0 and (i,j) in Preference\_set) or (WorkerFixed[i]=0 and WorkerOnlyPref[i]=0)) and x[i,j]>0.5} OUT

'ODBC'

'FileDSN=.\workforce\_allocation.dsn;'

'DROP TABLE IF EXISTS results\_x;'

'CREATE TABLE results\_x ( Worker INT, Location VARCHAR(1000));'

'results\_x' :

i ~ Worker, j ~ Location;

end;