|  |
| --- |
| **Knowledge Transfer Document**                      **Project Name: Trepicity**      **Date: March 03, 2017** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Trepicity**      **Knowledge Transfer Document** | | | | | |
|  | **NAME** | **SIGNATURE** | **DATE** | **COMPANY** | **JOB TITLE** |
| **Prepared By** | **Amit Jha** |  | 03-27-2017 | Prospus Consultancy Pvt Ltd |  |
| **Reviewed By** |  |  |  |  |  |
| **Approved By** |  |  |  |  |  |

**1.** **Introduction**

### 

### **Objective**

**Project Intend**

# **Solution**

# 

# 

# 

# 

# 

# 

# 

# **Overview of the Application**

|  |  |
| --- | --- |
| **Application Name** |  |

|  |  |
| --- | --- |
| **Portfolio / Domain** |  |
| **Team Member** |  |
| **Technology** |  |

**1**. **Purpose:**

**2**. **Scope:**

**3. Website:**

**1.**  **Module:**

|  |  |
| --- | --- |
| **PHP Files** |  |
| **Script Files** |  |
| **Database Tables** |  |
| **Accessible Roles** |  |

**a. Overview :**

**b.** **Functionality:**

**Functionality Name:**

|  |
| --- |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**1.** **Introduction**

### 

### **Objective**

This document is intended to capture the understanding of the application “Trepicity” during the Knowledge Acquisition phase. The objective is for Practice / Application support group Owners to assess the understanding, comfort level and effectiveness of the Knowledge acquisition phase and decide the readiness of proceeding to subsequent phases in the Knowledge acquisition phase gate review.

**Project Intend**

# **Solution**

The solution is a robust, comprehensive ‘vesselwise’ system built with open source platform PHP framework based architecture components. The system will facilitate straight-through processing all the way from authoring to compliance approval process to publication & distribution of the documents to various subscribers with a built-in automated workflow for efficiency.

# **Overview of the Application**

|  |  |
| --- | --- |
| **Application Name** | Trepicity WEBSITE |

|  |  |
| --- | --- |
| **Portfolio / Domain** | Trepicity.com |
| **Team Member** | Amit Jha, Shanti Gola, Awdhesh Soni, Saurabh Kumar, Mohit Kharbanda, Vaishali Shrivastava, Tarun Kumar |
| **Technology** | Cake PHP, MySql, jQuery. |

**1**. **Purpose:** The purpose of the Knowledge and Skills Transfer Strategy include:

Establish a strategy for the transfer of knowledge and skills related to the

technology and development done.

b. Share the knowledge of code and database implementation.

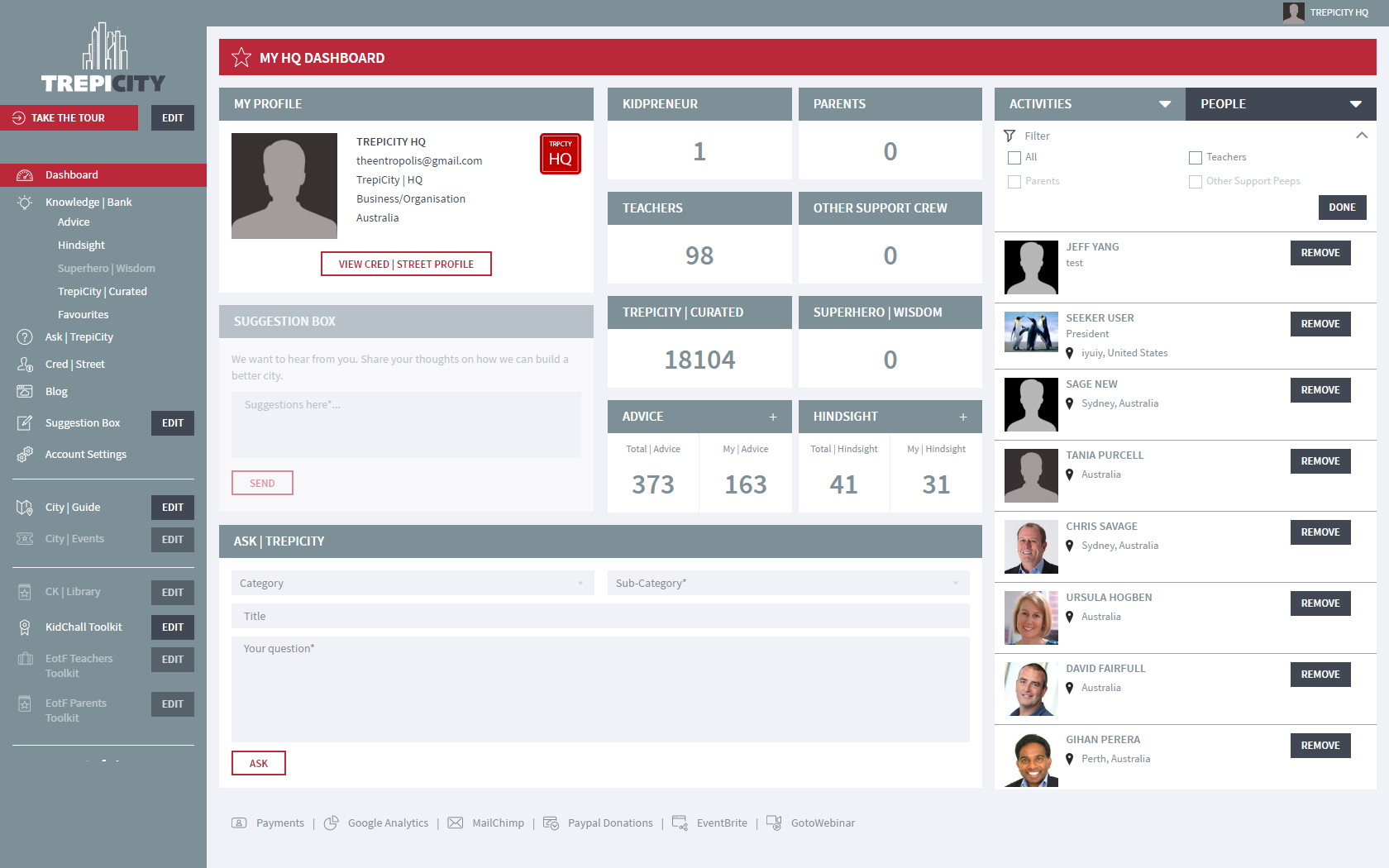
**2**. **Scope:**

**3.** **Website:**

**1.** **Dashboard Module:**

|  |  |
| --- | --- |
| **PHP Files** | Advices/dashboard.ctp  AdvicesController.php  Model/Invitation.php  activity\_notification\_element.ctp  people\_invitation\_element.ctp  advice\_all\_modal\_element.ctp  hindsight\_all\_modal\_element.ctp  blog\_js\_element.ctp  add\_hindsight\_js\_element.ctp  dashboard\_js\_element.ctp |
| **Script Files** | Script.js  Formmodule.js |
| **Database Tables** | |  | | --- | | Users  Students  Context\_role\_users  Publications  Advices  Hindsights | | Invitations  invitation\_status | | article\_published\_notifications | | Endorsements | | discussions,  question\_post\_likes  question\_post\_comments | | Suggestions  suggestion\_post\_views | | broadcast\_messages | | Invitations  Students | |
| **Accessible Roles** | Any users(Currently HQ and Teacher) |

1. **Overview:** Foradmin users,
   1. A user can see his/her profile in section “*My Profile*”.
   2. In central part user can see the counting of various objects such as types of users (Kidpreneur, Parents, Teachers, Other support peeps) and types of article like Advices, Trepicity | Curated, Hindsight. In article user can find the counting of that type of article counting included posted by him also.
   3. Ask | Trepicity: In Ask | Trepicity, user can ask any question to HQ.
   4. In right panel, there are two section activities and people. In activities user can various activity done globally by any user or done by the user who is in network. In People tab, the entire user in network is listed.



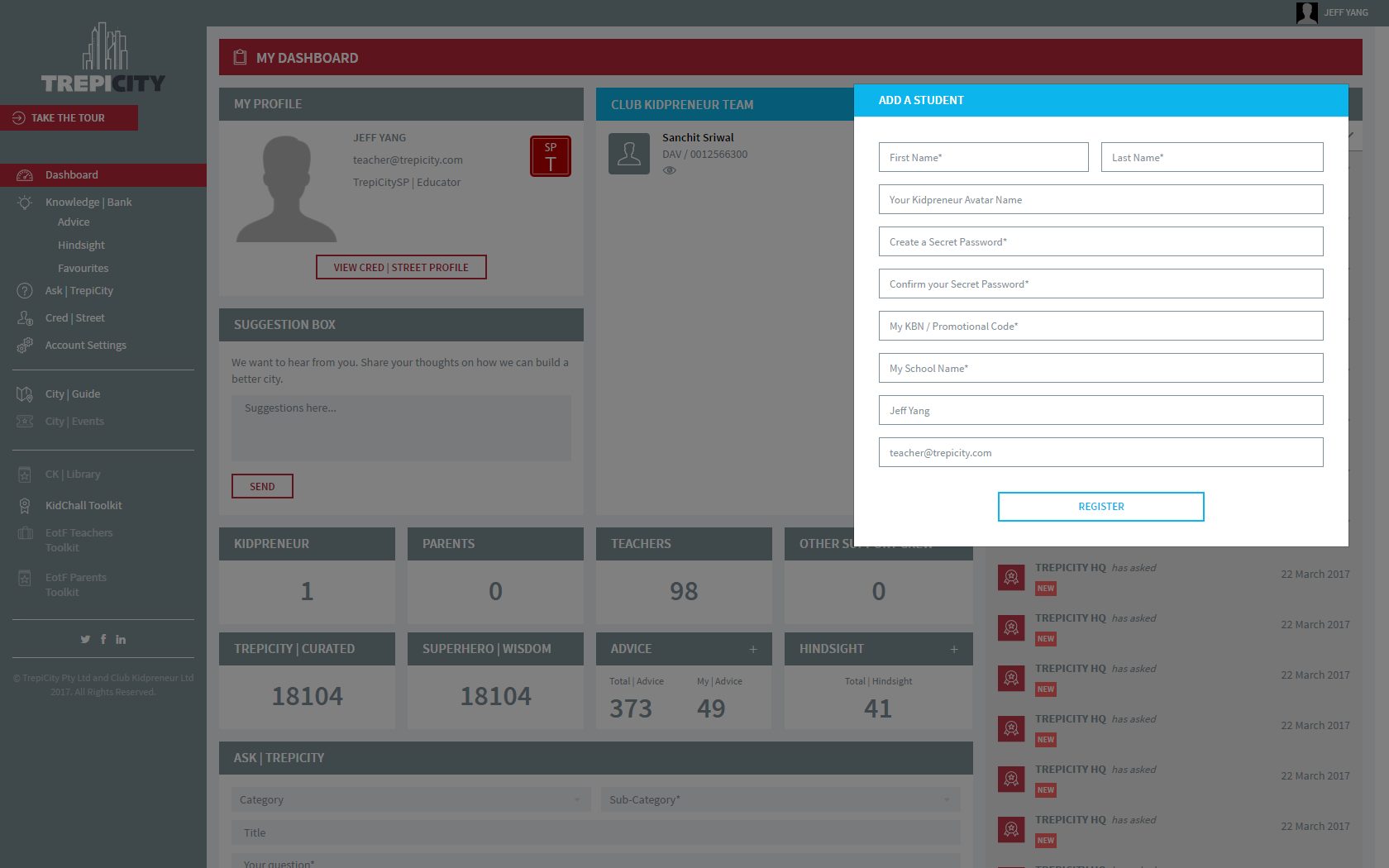
**b.** **Functionality:**

**Add Advice:** This functionality is same as the earlier version except the color and view.

**Add Hindsight:** This functionality is same as the earlier version except the color and view.

**Add Student**

* The students registered by the teacher should appear in their student list.
* When we click on Register button after adding the student details in the 'Add student' flyout
* For MVP we need to collect the Student details including their school and the teacher who registered them and email the detail to Trepicity HQ.
* We email the collected data to Trepicity HQ and also store in the DB for Phase 2 task.
* Check the form fields for the Kidpreneur signup form (in Invision) to set up the database
* In the Add student flyin for adding students, the teacher does not add any user image for the students. This be taken care of in Phase 2 when we will work on Kidpreneur settings page. We will keep this as a grey box for now in MVP when a student is added.
* The following fields are shown in the student's detail on the Teacher dashboard once a student is added.
* [First Name] [Last Name]
* [School Name] [KBN]
* When a student is added/removed, a Confirmation should be sent to the Teacher only
* Entropolis had the same counter ‘system’
* Each time we add a new article to the TrepiCity|Curated the counter adds +1 to the total
* Same for Superhero|Wisdom;
* Each time a Teacher signs up the counter add +1
* Same for Other Support Crew
* Each time a Teacher registers a student via their dashboard the Kidpreneur counter adds +1
* For Phase 2 we will use this data to pre-populate the Kidpreneur database and set up the Kidpreneur dashboards when available.



To add new student, ajax is used to call method defined in user model. In user model, query is written for new student and then response a message for user. Same methodology is applied for edit student.

**2.** **Take the tour:**

|  |  |
| --- | --- |
| **PHP Files** | tourVideosController.php  tourVideo.php  tourVIdeos/index.ctp  tourVIdeos/add.ctp  tourVIdeos/edit.ctp  tourVIdeos/view.ctp |
| **Database Tables** | |  | | --- | | tour\_videos  tags | |
| **Accessible Roles** | HQ to add/edit, teacher for view only. |

Requirement:

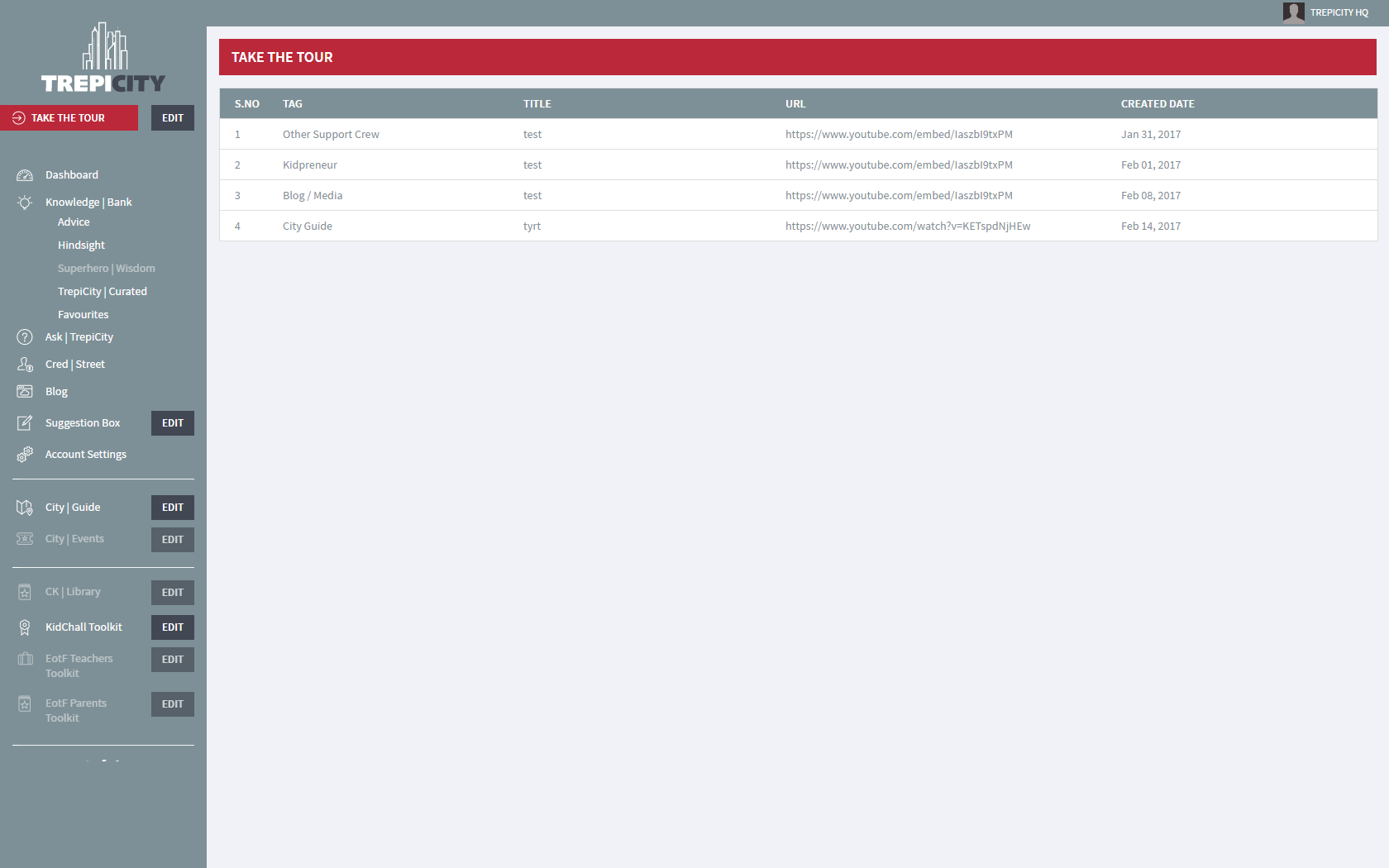
This is a link to a demo video for teacher.

Load a page when the TRPCITY HQ clicks on the Take the tour menu Item with a list like where we can upload the videos / edit / delete.

Edit: When the HQ clicks on the EDIT button, the Add video and edit and delete actions will appear.

Edit (Take the Tour)

Trepicity HQ should be able to delete videos and upload new videos and tag for each user type (currently only Teacher)



Functionality : functionality is based on basic Cake php standard.

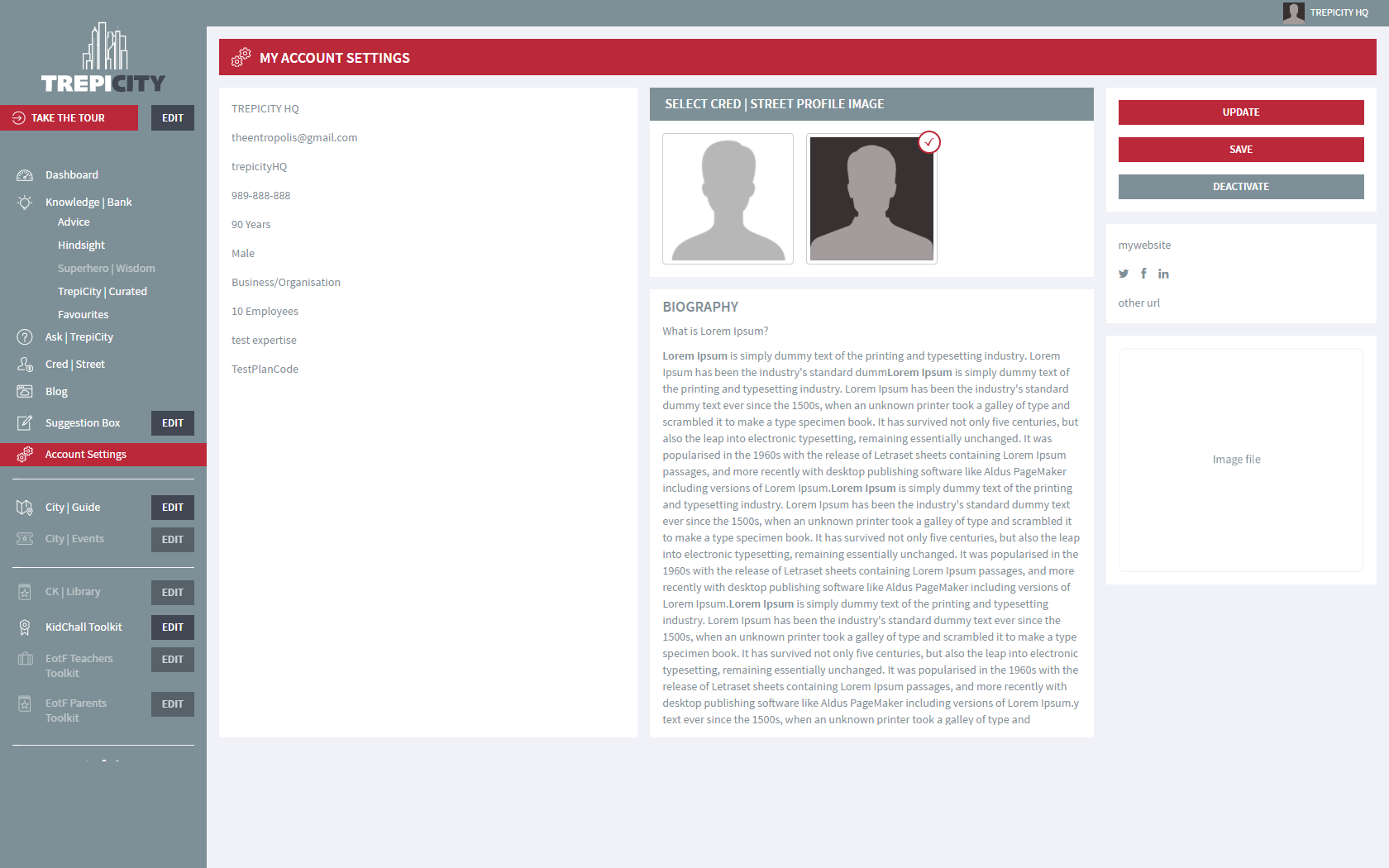
**Account Setting**

|  |  |
| --- | --- |
| **PHP Files** | Controller/usersController.php  model/user.php  View/users/profile.ctp  View/users/setting.ctp |
| **Database Tables** | |  | | --- | | users  user\_profiles | |
| **Accessible Roles** | Accessible by all users. |

* The Account settings page from ENTR v1.0 is replaced by Support Crew Settings.
* The Support Crew Settings will pre-populate with mandatory information from the Support Crew registration form
* The User can update with further information via the EDIT button on the support Crew Settings Page
* The User can deactivate their account using the DEACTIVATE button
* When user deactivates please send alert to ENTR HQ
* There should be a communication workflow attached to the deactivation the same as in ENTR
* The pre-defined dropdown options for the following fields will be the ones from ENTR for now. Tania will double check when she will review the ENTR site properly
* Age
* Gender
* Area of Interest and/or expertise
* Size of Organization
* Add another field for ‘Blog’ in the Account settings. This is for the user to enter their own blog address if they have one. This is a non-mandatory field. Map this field with the Support crew profile flyin Blog link.
* In ‘SELECT CRED | STREET PROFILE IMAGE’, the user can upload 3 profiles images. This should match the functionality to ENTR. This functionality has not changed for Trepicity
* Note: this page is the same on all Support Crew dashboards including Teacher Dashboard
* HQ will see the same screen as all other users and will be able to update their own details that appear in the Cred|Street alongside all other users
* Users including HQ can edit their existing details in here or complete any pending detail.
* Users can deactivate their own account from this section. HQ should be alerted when an account is deactivated. There should be a communication workflow attached to the deactivation the same as in ENTR
  + After the users deactivate their account from the Account settings, They will have to re-register
    - Once they deactivate their account, There should be a pop-up that says
      * ARE YOU SURE YOU WANT TO DEACTIVATE YOUR ACCOUNT?
      * YES and NO buttons
      * Once they select YES button there should be another pop-up that says
      * YOUR TREPICITY ACCOUNT IS NOW DEACTIVATED. SORRY TO SEE YOU GO. IF YOU WOULD LIKE TO REACTIVATE YOUR ACCOUNT IN THE FUTURE PLEASE RE-REGISTER.
      * OK button
    - Once user has deactivated their account an alert should be sent to HQ@trepicity.com
  + After deactivation, Only view access to the website public pages on the URL
* For the following pre-defined dropdown options, use the ones from ENTR for now. Client will double check when they will review the ENTR site properly
  + Age
  + Gender
  + Area of Interest and/or expertise
  + Size of Organization
  + Update is same as EDIT. SAVE is for saving edits.
  + For the ‘image file’ location of the page, this will be a business logo image or marketing meme to represent their business.

Note: This is similar to the functionality that is available on LinkedIn but this requirement can be moved to Phase 2 / 3 as low priority.

We will build the full functionality including HQ in Phase 2



Functionality:

View Profile: Default page is of account setting is View page. Here user can update profile picture only. For edit mode user has to click on Update button and then can update profile related details.

**Blog/Media Page**

|  |  |
| --- | --- |
| **PHP Files** | Controller/advicesController.php  Controller/pagesController.php  model/advice.php  View/Pages/blog.ctp  View/TourVideos/add.ctp |
| **Database Tables** | |  | | --- | | Advices, tourvideos | |
| **Accessible Roles** | Blog page can be accessible publicly.  Only HQ can add details of blog. |

* On the HQ Dashboard, when we will click on the Blog action from the left menu bar, it will open this Screen/popup through which the Blogs can be uploaded. The link should re-direct you to this screen not the website page
  + Style update is to Change the yellow to red.
* Functional updates:
  + Able to ‘tag’ different Blog / Media types and store them so they can be viewed from the website blog page.
    - We just need to be able to upload pdfs and .mov, select a category / sub-category that is relevant to the file and then select the TYPE so they appear in the correct section
    - Tagging and Blog / Media is a HQ admin functionality only
  + We need to add a ‘type’ tag functionality the PUBLISH NEW ADVICE screen to ensure they appear in the correct list on the BLOG / MEDIA Page and can be viewed by anyone visiting the website
  + For HQ we would have the following options as Tag list in dropdown. Tagging will ‘send’ the blog media to the correct section of the Blog / Media page so they are catalogued according to type:
    - Advice (this goes straight to Knowledge | Bank marked as Advice)
    - Feature
    - Trepicity HQ
    - Club Kidpreneur HQ
    - Latest Posts
    - Archived Posts
    - Media and PR
    - Videos
  + All other Support Crew will use the standard Advice | Market functionality no change other than design to the ENTR v1.0 Advice | Market
* On the HQ Dashboard, when we will click on the Blog action from the left menu bar, it will open this Screen/popup through which the Blogs can be uploaded. The link should re-direct you to this screen not the website page
  + Style update is to Change the yellow to red.

**Publish Organization:** SA has right to publish an organization, this make

organization approved and make available to Organization Administrator. Only  
 those Vessel Manufacturer are available whose corresponding organization are

approved. In organization table, status column value is set to 1 and mail is send

to all SA with the information about this publish organization.

**Merge Organization:** While merging a organization, a list of approved

organization is available as shown in below image. After select one of these

approved organization and click merge action. Current organization is merged

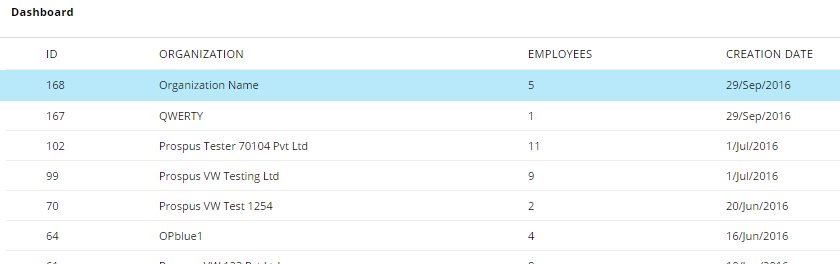
with select organization and this organization organization is unavailable. The

merge\_id column of current organization has value of select organization and its

corresponding vessel manufacturer is unavailable and mail is send to all SA with

the information about this merge organization. Below is picture of available

organization for merge



**2.** **Organization Module:**

|  |  |
| --- | --- |
| **PHP Files** | dashboard.php  /ajaxhandler/organization-ajaxhandler.php  /sections/organization.php  class.organization.php  class.pagination.php  class.user.php |
| **Script Files** | /js/organization.js  /js/pagination.js |
| **Database Tables** | organization  organization\_category  organization\_category\_join  country  state |
| **Accessible Roles** | System Administrator, Organization Administrator |

**a. Overview :** SA and OA have access to this section. SA can Add, edit, publish

and merge to all Organization whereas OA can edit, publish and merge only its

own organization. For System Administrator role, there is unapproved

organization section. SA can edit, merge and publish select organization and a

mail is send to all SA, organization administrator. In merge organization, current

select organization is merged with approved organization and its corresponding

vessel manufacturer organization is updated. In publish organization, it is directly

published but its other detail remains empty. Below is organization category image and organization\_category table structure.

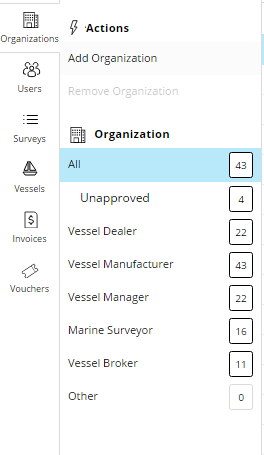
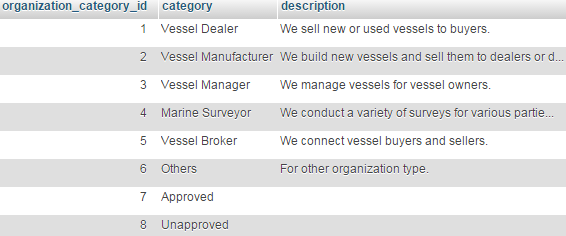


Image. Organization Category



**b.** **Functionality:**

**Add and Edit Organization:** SA can add or edit any organizations but OA can

only edit his own organization.

**Publish Organization:** SA has right to publish an organization, this make

organization approved and make available to Organization Administrator. Only  
 those Vessel Manufacturer are available whose corresponding organization are

approved. In organization table, status column value is set to 1 and mail is send

to all SA with the information about this publish organization.



**Merge Organization:** While merging a organization, a list of approved

organization is available as shown in below image. After select one of these

approved organization and click merge action. Current organization is merged

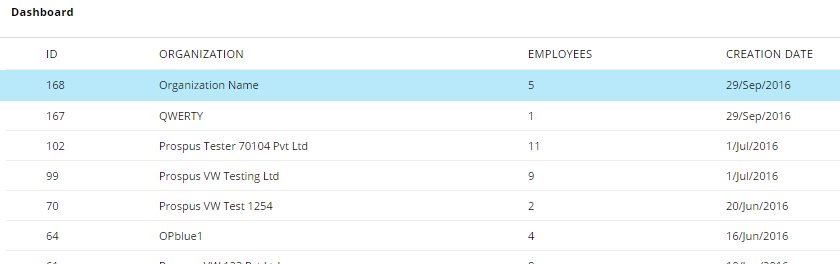
with select organization and this organization organization is unavailable. The

merge\_id column of current organization has value of select organization and its

corresponding vessel manufacturer is unavailable and mail is send to all SA with

the information about this merge organization. Below is picture of available

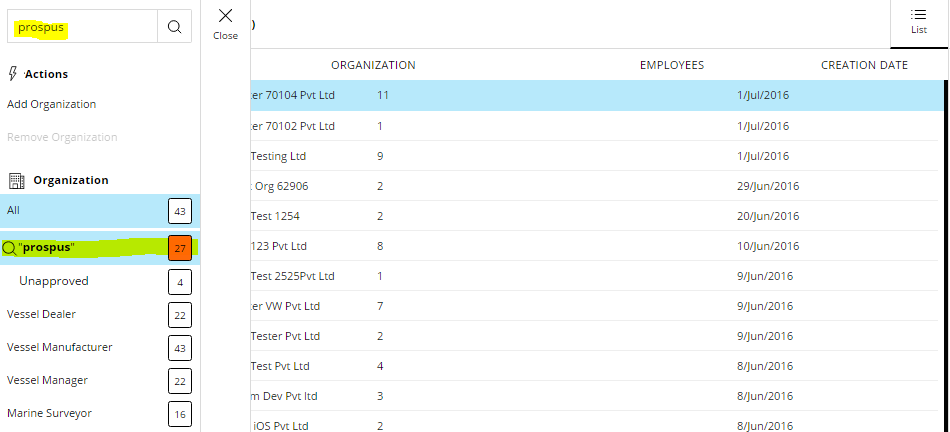
organization for merge



**Search Organization:** Organization search details on organization name,

address, website, phone, employee, organization category. User has edit,

publish, merge functionality for organization.



**3. Survey Module:**

|  |  |
| --- | --- |
| **PHP Files** | dashboard.php  /ajaxhandler/item-ajaxhandler.php  /ajaxhandler/item-ajaxhandler-extended.php  /sections/survey.php  /template/item\_history\_view\_template.php  class.item.php  class.survey  class.pagination.php |
| **Script Files** | /js/item.js  /js/item-extended.js  /js/survey.js  /js/pagination.js |
| **Database Tables** | item  item\_image  item\_history  item\_image\_history  Item\_priority\_value  item\_priorit\_value\_history |
| **Accessible Roles** | System Administrator  Organization Administrator  Vessel Owner  Vessel Administrator  Vessel Crew  Vessel Guest  Observer |

**a. Overview :** In this section, i have work on item history. In item history when a

user edit item then its history is stored in database. Item history detail in attach to

item detail in item view mode. Below is picture of a item history.

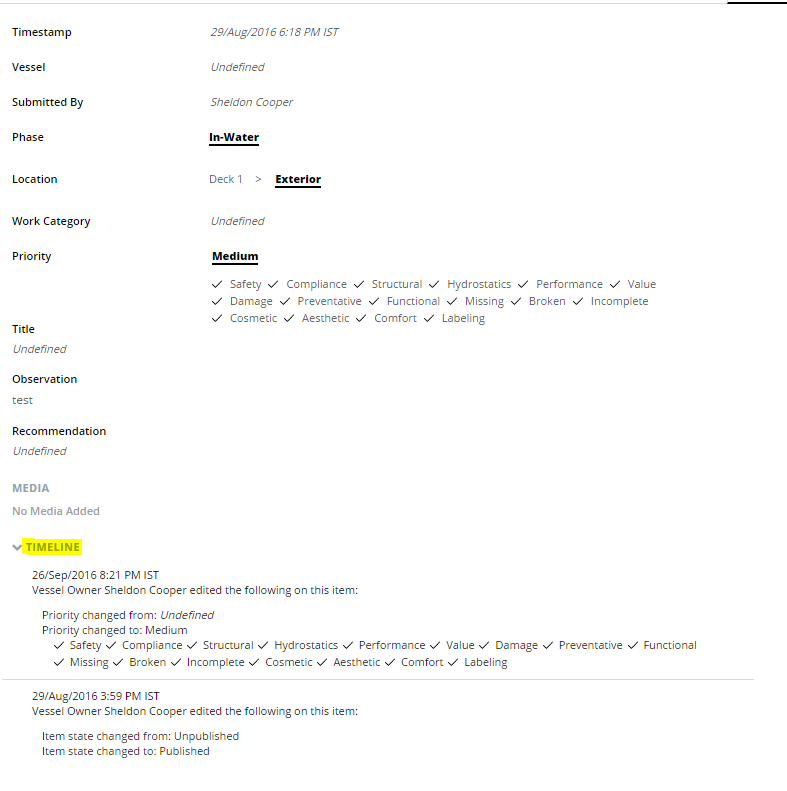


Image. Item Timeline.

**b. Functionality:**

**Item History:** When user publish or edit an item, it’s current value store in item and item\_image tables is copy to item\_history and item\_image\_history tables respectively. Thereafter item and item\_image tables are updated with new data. Vessel area of the item is stored in vessel\_area field in text form instead of id.

In item view mode, current item table value is compared with last entry of same

item in item\_history table. If any difference is there then a single item history is

generated, now after that we compare item\_history table consecutive

item\_history\_id in decrease order for current item\_id. In this way item history is created.

**4. Vessel Module:**

|  |  |
| --- | --- |
| **PHP Files** | dashboard.php  /ajaxhandler/dashboard\_vessel\_ajaxhandler.php  /sections/vessel.php  class.vessel.php  class.pagination.php |
| **Script Files** | /js/vessel.js  /js/pagination.js |
| **Database Tables** | vessel  vessel\_manufacturer  organization |
| **Accessible Roles** | System Administrator  Organization Administrator  Vessel Owner  Vessel Administrator  Vessel Crew  Vessel Guest |

**a. Overview :** In this section, i have work on vessel manufacturer auto suggestion.

**b. Functionality:**

**Vessel Manufacturer Auto Suggest:** On type in vessel manufacturer input on add

vessel or edit vessel. A auto suggestion of vessel manufacturer is display from

which user can select if he is try to enter already exist vessel manufacturer. Each

manufacturer has corresponding organization, only those vessel manufacture

comes in suggestion whose corresponding organization are approved.

**5. Global Module:**

|  |  |
| --- | --- |
| **PHP Files** | dashboard.php  class.pagination.php |
| **Script Files** | /js/validetta.js  /js/pagination.js |
| **Accessible Roles** | All role |

**a. Overview :** Two global implementation are done which include Global Validation

and Global Pagination.

**b. Functionality:**

**Global Validation:** While submit any form in any section of the website after login.

If form field validetta is required then validation with input field border color red is

done by adding class inputfielderror to text type, radiofielderror to

radio type, checkboxfielderror to checkbox type.



Image. Add inputfielderror class to the input type text.



Image. Add radiofielderror class to the input type radio.

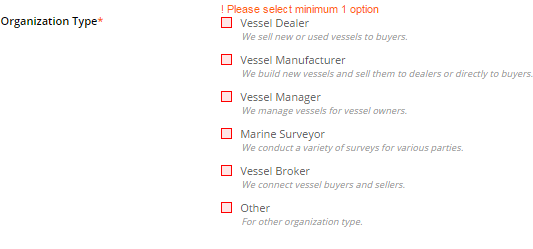


Image. Add checkboxfielderror class to the input type checkbox.

**Global Pagination Option Bar Button:** By default 100 rows of data is shown in

pagination of survey, user, vessel, organization. By click leftmost button on

pagination section, user can get all data at once. For that this button has

Button-data, onclick this nextfn function is call on pagination.js page with

required parameter.This is consider as Bar Button.

Image left most yellow color button is Bar Button, used to show all data at once.

**Global Pagination Option Power Button:** By default 100 rows of data is shown

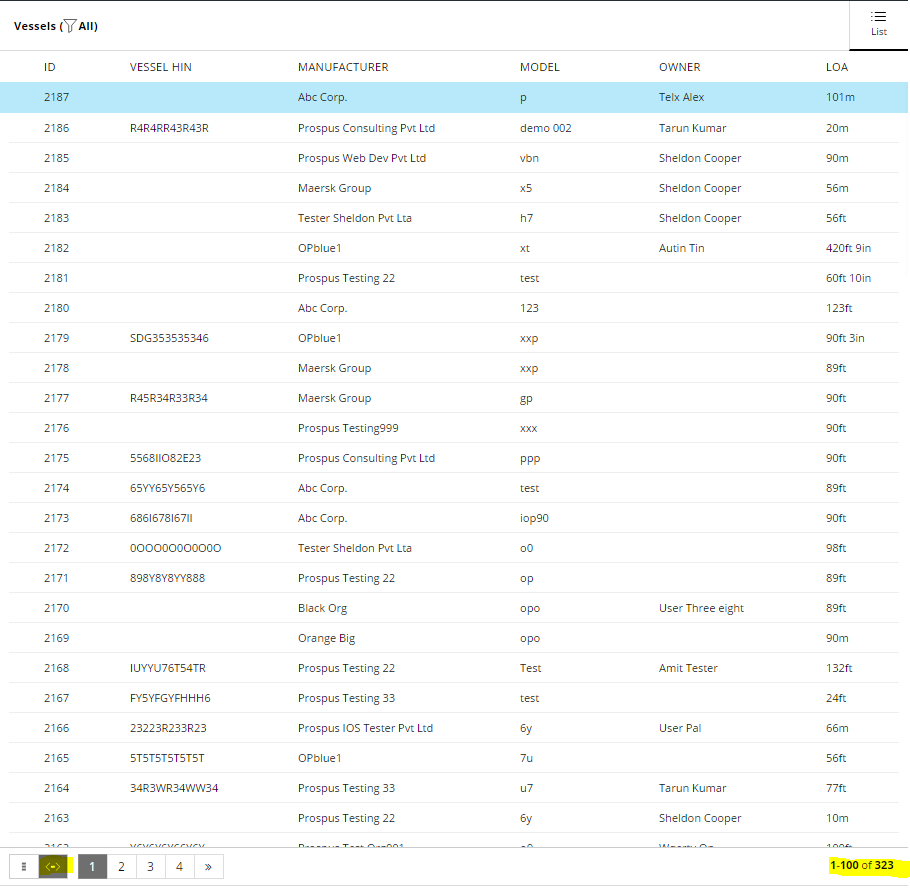
in pagination of survey, user, vessel, organization. By click leftmost button on

pagination section, user can get 100 rows of data. If user trigger Bar button now

if he want to show only 100 rows of data, user can click this button. For that this

button has Button-data, onclick this nextfn function is call on pagination.js

page with required parameter.This is consider as Power Button.

Image left yellow color button is Power Button, used to show 100 data at once.

|  |  |
| --- | --- |
| **Portfolio / Domain** | Web Service |
| **Team Member** | Shaan Gola, Manish Tomar, Amit Bhardwaj. |
| **Technology** | PHP, json, MySql. |

**1**. **Purpose:**

1. VesselWise for devices works for 5 roles which include Vessel Owner, Vessel Administrator, Vessel Crew, Vessel Guest, Observer.

b. User has access to Dashboard and Survey. Main purpose of this app is to provide

user access to these section in online and offline mode.

**2**. **Scope:**

1. VesselWise brings a systematized, itemized, organized and optimized workflow-based solution for various vessel-stakeholders. With Vessel Owner, Vessel Administrator, Vessel Crew, Vessel Guest, Observer have access to.

b. Complete list of survey items available under one, centralized repository.

**3.** **WebServices-**

A single file ws.php has all code for web services with some include file. A device build

have a Web Services Version such that other builds doesn’t affect it. Following are

functionality defined in web services file.

These are two data that are frequently used in web services.

a. Lastlogin request parameter is used for last sync time for the device. If this is sent

for any web services then data will be send which are update after this time.

b. Updating\_device request value will be (1 => ipad, 2 => iphone, 3 => android). It is

use to know request device.

c. DEVICE\_UUID request value is unique value for the device. It is sent from the

device and store to database while saving an item, chat etc.

**1. User Login:** loginUserMethodIpd switch case use for user login and call function loginUserMethodIpd with parameter user email, password, device (1 => ipad, 2 => iphone, 3 => android), device token (unique for each device, use for chat notification). Return json has user id with details , user role , success (1=> for success login, 0=> for unsuccessful login), code ( 200 => for wrong credential, 205 => only check profile is completed, if profile is not completed code will be 205, 206 => if user is registered and eligible to login but not assign any general survey 206, 207 => if user subscription is ended).

|  |  |
| --- | --- |
| **Database Tables** | user  user\_role  role  User\_role\_invitation  Vessel |
| **Switch Case** | loginUserMethodIpd |

**2.** **Verify Connection with Server:** return json for checking connection with server.

|  |  |
| --- | --- |
| **Switch Case** | verifyNetConnection |

**3.** **Get Roles:** GetRoles switch case use for sending available roles for the device which are include Vessel Owner, Vessel Administrator, Vessel Crew, Vessel Guest, Observer, and Vesselwise Guest.

|  |  |
| --- | --- |
| **Database Tables** | role |
| **Switch Case** | GetRoles |

**4.** **Vessel Manufacturer:** GetVesselMf switch case use for getting vessel manufacturer and call function getVesselMf\_all, getVesselMf\_all\_Sync with parameter device (1 => ipad, 2 => iphone, 3 => android) and Lastlogin.

|  |  |
| --- | --- |
| **Database Tables** | vessel\_manufacturer |
| **Switch Case** | GetVesselMf |

**5.** **Role Invitation Accept/Reject:** SetRoleInvitationByUser switch case use to update invitation on a vessel. In request json, updating is done for user\_role\_invite\_id.

|  |  |
| --- | --- |
| **Database Tables** | user\_role\_invitation, user\_role, survey, vessel\_area, item |
| **Switch Case** | SetRoleInvitationByUser |

**6.** **Survey List:** GetSurveyList switch case use to return survey, item, and item history for the login user. It works base on Lastlogin date time. If Lastlogin is zero then all above data will be sending. If a date time is given then after that date time data will be send.

|  |  |
| --- | --- |
| **Database Tables** | survey, vessel, survey\_user, vessel\_manufacturer,  vessel\_model, survey\_type, user\_role\_invitation, item, item\_image, item\_history, item\_image\_history, priority\_value, item\_priority\_value\_history |
| **Switch Case** | GetSurveyList |

**7.** **Vessel Area:** GetVesselArea switch case use to get vessel area of vessel to which current user is either Accept or Pending. This also works for Lastlogin.

|  |  |
| --- | --- |
| **Database Tables** | user\_role\_invitation, vessel\_area |
| **Switch Case** | GetVesselArea |

**8.** **User detail by id:** GetUsersById switch case use to get user information from request user id. The response json has user id with details, user role.

|  |  |
| --- | --- |
| **Database Tables** | role, user, user\_role, user\_role\_invitation, item\_user |
| **Switch Case** | GetUsersById |

**9.** **Chat for an item:** GetFileChatListByRoomId switch case use to get chat from request chat room id. Input json has device local chat id, so this id and DEVICE\_UUID pair is used to make a single chat unique. Response json could have text and image both. Chat text will be sending in encode form. Response json also include chat on the chat room after the Lastlogin date time. It also includes Item history.

|  |  |
| --- | --- |
| **Chat Text File Location** | /ajaxhandler/chatlog/ |
| **Database Tables** | role, chat\_room, user, user\_role,item\_image, item\_history, item\_image\_history, priority\_value, item\_priority\_value\_history |
| **Switch Case** | GetFileChatListByRoomId |

**10.** **Forget Password:** ForgetPasswordEmail switch case use to reset password. Email Id and code is input from the device. This code is sending in mail to the email id.

|  |  |
| --- | --- |
| **Database Tables** | user, user\_role, role, user\_profile |
| **Switch Case** | ForgetPasswordEmail |

**11.** **Change Password:** ChangeUserPassword switch case use by device user after Forget Password service. User email id and new password are input from device in this case.

|  |  |
| --- | --- |
| **Database Tables** | user |
| **Switch Case** | ChangeUserPassword |

**12.** **Save chat in file:** SetFileChatList switch case use by device for dialogue on an item. Input json have all the information about the chat (chat\_room\_id, message, sender\_id, sender\_role) with device\_chat\_id which is stored in file with DEVICE\_UUID such that if this service use again with same device\_item\_id and DEVICE\_UUID then these field will be match and return previous entry detail. Response json could have text and image both. Chat text will be sending in encode form. Response json also include chat on the chat room after the Lastlogin date time. It also includes Item history.

|  |  |
| --- | --- |
| **Database Tables** | Chat\_room, chat\_user\_room, item, user, vessel\_sub\_role, vessel, survey, item\_history, item\_image\_history, priority\_value, item\_priority\_value\_history |
| **Switch Case** | SetFileChatList |

**13.** **Save Item Image:** SaveImageItem switch case use for uploading a new image. Input json have device\_item\_image\_id then its corresponding item\_image\_id is sent in response json.

|  |  |
| --- | --- |
| **Database Tables** | item, item\_image |
| **Switch Case** | SaveImageItem |

**14.** **Update Item Image:** SaveImageItemById switch case use for replace item image with input json item\_image\_id.

|  |  |
| --- | --- |
| **Database Tables** | Item\_image, item |
| **Switch Case** | SaveImageItemById |

**15.** **Login user vessel:** GetVesselByUserDetail switch case return all vessel for which current login is assigned role from Vessel Owner, Vessel Administrator, Vessel Crew, Vessel Guest, and Observer. It returns other user role detail for above vessel. It uses two table user\_role\_invitation and user\_role\_invitation\_deleted. When a user role is remove or reject it user\_role\_invitation entry move to user\_role\_invitation\_deleted table through trigger. Entry from both tables is send to the user.

|  |  |
| --- | --- |
| **Database Tables** | user, user\_role\_invitation, role, vessel, user\_role\_invitation\_deleted, survey, vessel\_sub\_role, item\_user, organizations, item\_user |
| **Switch Case** | GetVesselByUserDetail |

**16.** **Update or New Vessel Area:** SetVesselAreaList switch case use to update or delete vessel area while editing an item. It update last\_updated\_time field of vessel\_area table, so that this updated data to be sent.

|  |  |
| --- | --- |
| **Database Tables** | vessel\_area |
| **Switch Case** | SetVesselAreaList |

**17.** **Flag Status Change:** changeFlagStatus switch case use to change item and vessel\_area flag field from 0 to 1. When an item or vessel\_area create initially its flag field is 0, so it is inaccessible and its corresponding id is sent to device. Using that id this flag is converted to 1 and available.

|  |  |
| --- | --- |
| **Database Tables** | Item, vessel\_area |
| **Switch Case** | changeFlagStatus |

**18.** **Save Survey:** SetSurveyList switch case use to update or create an item. Input json with change\_type field used for either new item or update the existing item. For new change\_type device \_item\_id with device unique id. If same request is send to server again then web services verify if entry already done and its return json include database item\_id. Each item section has item history subsection. Item history is saved in item\_history table. Response json has system item\_id and history of the item.

|  |  |
| --- | --- |
| **Database Tables** | Item, item\_priority\_value, item\_image |
| **Switch Case** | SetSurveyList |

**19.** **Role Waiting Invitation:** GetRoleWaitingInvitationDetail switch cases return all role invitation to the user or by the current login user. This detail is shown on dashboard section. Return json divided in two sections me and other, me has current login user invitation and other has invitation send by the user.

|  |  |
| --- | --- |
| **Database Tables** | user, user\_role, user\_role\_invitation, vessel, organization\_user\_role, context |
| **Switch Case** | GetRoleWaitingInvitationDetail |

**20.** **Waiting Dialogue:** GetWaitingDialoguesDetail switch cases return all dialogue waiting to the user or by the current login user. This detail is shown on dashboard section. Return json divided in two sections me and other, me has current login dialogue waiting and other has dialogue send by the user.

|  |  |
| --- | --- |
| **Database Tables** | item\_user, role, user\_role, item\_user, chat\_user\_room, item |
| **Switch Case** | GetWaitingDialoguesDetail |

**21.** **User Status by email:** GetUserStatusByEmail switch cases return user and its detail. Input json has id corresponding to which response json has code. Code : 1=> no record, 2 => inactive, 3 => active , 4 => password change on active account, cancel => 5, 6 => invited, 7 => rejected, 8 => expired.

|  |  |
| --- | --- |
| **Database Tables** | user |
| **Switch Case** | GetUserStatusByEmail |

**22.** **Item PDF report share:** ShareItemPdfReport switch cases share item pdf report. It shares multiple items to an active user. It assign Observer role and send an email with share pdf report.

|  |  |
| --- | --- |
| **Database Tables** | user, survey, vessel, vessel\_manufacturer, vessel\_model, user, user\_profile, survey\_user, user\_role, item, item\_user, survey\_type |
| **Switch Case** | ShareItemPdfReport |