

**BILKENT UNIVERSITY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**CS353**

**Database Systems**

**Final Report**

Social Network and Forum for Researchers

**Group 11**

**Group Members**

Ahmet Ay 21200701

Berk Bozkurt 21101098

Cüneyt Erem 21202398

Mert Gurcan, 21101646

**Instructor**

A. Ercüment Çiçek

Final E/R Diagram



**1. Project Describe**

Our project is the professional network for scientists and researchers. We help researchers connect and make it easy for them to share and access scientific output, knowledge, and expertise.

**2. Set of Relations**

There are attribute domains, candidate keys (and the selected, primary key), and foreign keys due to referential integrity was well as 3NF type.

**2.1 User**

**craete table** User(

userID **int** (8) **not null AUTO\_INCREMENT,**

username **varchar**(30) **not null**,

password **varchar**(30) **not null,**

email **varchar**(30) **not null,**

name **varchar**(30) **not null**,

age **int**(2) **not null,**

gender **varchar**(8) **not null,**

deptID **int** (8)**not null,**

instID **int** (8)**not null,**

**primary key** (userID ),

**foreign key** (deptID) **references** Department

**foreign key** (instID) **references** Institution

)

**Functional Dependencies:**

userID -> username , password, email

**Candidate key:**

**(**userID, username)

**Normal Form:** 3NF

---

**2.2 Author**

**create table** Author(

userID **int** (8) **not null,**

**primary key** (authorID ),

**foreign key** (userID) **references** User

**);**

**Functional Dependencies:**

authorID -> userID

**Candidate key:**

userID

**Normal Form:** 3NF

---

**2.3 Department**

**create table** Department(

deptID **int** (8) **not null AUTO\_INCREMENT,**

deptName **varchar(**30) **not null,**

**primary key(**deptID **);**

);

**Functional Dependencies:**

deptID -> deptName

**Candidate key:**

**none**

**Normal Form:** 3NF

---

**2.4 Institution**

**create table** Institution(

instID **int** (8) **not null AUTO\_INCREMENT,**

instName **varchar**(30) **not null,**

country **varchar**(30) **not null,**

**primary key(**instID **);**

**);**

**Functional Dependencies:**

instID -> instName , country,

**Candidate key:**

**(**instID,instName),instID

**Normal Form:** 3NF

---

**2.5 Post**

**create table** Post(

pID **int** (8) **not null AUTO\_INCREMENT,**

title **varchar**(30) **not null,**

dateCreated **varchar**(30) **not null,**

dateModified **varchar**(30),

userID **int** (8)

**primary key** (pID )

**foreign key** (userID) **references** User

);

**Functional Dependencies:**

pID -> title, dateCreated, dateModified

**Candidate key:**

pID

**Normal Form:** 3NF

**2.6 Article**

**create table** Article(

content **blob** **not null,**

pID **int** (8) **not null,**

topicName **varchar**(30) **not null,**

**primary key(**pID **);**

**foreign key** (pID) **references** Post

**foreign key** (topicName) **references** topic

);

**Functional Dependencies:**

pID -> content

**Candidate key:**

pID

**Normal Form:** 3NF

---

**2.7 Question**

**create table** Question(

qArea **varchar**(30) **not null,**

pID **int** (8) **not null,**

**primary key(**pID **);**

**foreign key** (pID) **references** Post

);

**Functional Dependencies:**

pID -> qArea

**Candidate key:**

pID

**Normal Form:** 3NF

---

**2.8 Topic**

**create table** Topic(

topicName **varchar**(30) **not null,**

**primary key(**topicName **);**

);

**Functional Dependencies:**

none

**Candidate key:**

topicName

**Normal Form:** 3NF

---

**2.9 Tag**

**create table** Tag(

tagName **varchar**(30) **not null,**

**primary key(**tagName **);**

);

**Functional Dependencies:**

none

**Candidate key:**

tagName

**Normal Form:** 3NF

---

**2.10 Comment**

**create table** Comment(

pID **int** (8) **not null**

cArea **varchar(1000) no null,**

index **int** (8) **not null,**

**primary key(**index, pID**);**

**foreign key** (pID) **references** Post

);

**Functional Dependencies:**

none

**Candidate key:**

index, pID

**Normal Form:** 1NF

---

**2.11 writes**

**create table** writes(

authorID **int** (8) **not null,**

pID **int** (8) **not null,**

**primary key(**authorID,pID **);**

**foreign key** (authorID ) **references** Author

**foreign key** (pID ) **references** Post

);

**Functional Dependencies:**

none

**Candidate key:**

pID,authorID

**Normal Form:** 3NF

---

**2.12 likes**

**create table** likes(

userID **int** (8) **not null,**

pID **int** (8) **not null,**

**primary key(**userID,pID **);**

**foreign key** (userID ) **references** User

**foreign key** (pID ) **references** Post

);

**Functional Dependencies:**

none

**Candidate key:**

userID,pID

**Normal Form:** 3NF

---

**2.13 favorite**

**create table** favorite(

userID **int** (8) **not null,**

topicName **varchar**(30) **not null,**

**primary key**(userID);

**foreign key** (userID ) **references** User

**foreign key** (topicName ) **references** Topic

);

**Functional Dependencies:**

none

**Candidate key:**

userID

**Normal Form:** 3NF

---

**2.14 hastag**

**create table** hastag(

pID **int** (8) **not null,**

tagName **varchar**(30) **not null**

**primary key**(pID);

**foreign key** (tagName ) **references** Tag

);

**Functional Dependencies:**

none

**Candidate key:**

pID

**Normal Form:** 3NF

**2.15 follow**

**create table** follow(

followerID **int**(8) **not null,**

targetID **int(8) not null,**

**primary key**(followerID, targetID );

**foreign key** (targetID ) **references** User

**foreign key** (followerID ) **references** User

)

**Functional Dependencies:**

none

**Candidate key:**

(followerID, targetID )

**Normal Form:** 3NF

---

**2.16 reference**

**createn table** follow(

refID **int**(8) **not null,**

targetrefID **int(8) not null,**

**primary key**(refID, targetrefID );

**foreign key** (refID ) **references** Article

**foreign key** (targetrefID ) **references** Article

)

**Functional Dependencies:**

none

**Candidate key:**

(refID, targetrefID )

**Normal Form:** 3NF

---

**2.17 foreignReferences**

**createn table** foreignReferences(

foreignRefID **int**(8) **not null,**

foreignRefAuthorName **varchar**(30) **not null,**

foreignRefTitle **varchar**(30) **not null,**

foreignReferenceURL **varchar**(200) **not null,**

pID **int** (8) **not null**

**primary key**(foreignRefID );

**foreign key** (pID ) **references** Post

);

**Functional Dependencies:**

foreignRefID ->foreignRefAuthorName,foreignRefTitle,foreignReferenceURL

**Candidate key:**

foreignRefID,(foreignRefAuthorName,foreignRefTitle)

**Normal Form:** 3NF

3. **Implementation Details**

- Our project is based on xhtml and java. With sql statement, we connect database, get information from there.In order to use it on xhtml, we store it as a java object. User entrance ,such as post question, will send to the database. Related sql is given below.

Following codes are for inserting new question

PreparedStatement preparedStatement = connection.prepareStatement("insert into post (title, dateCreated, dateModified, userID) VALUES (?, ?, ?, ?)");

preparedStatement.setString(1, questionTitle);

preparedStatement.setString(2, dateCreated);

preparedStatement.setString(3, dateCreated);

preparedStatement.setInt(4, userID);

Following codes are inserting new user

preparedStatement = connection.prepareStatement("INSERT INTO `user\_table`(`username`, `password`, `email`, `name`, `age`, `gender`) "

+ "VALUES (?,?,?,?,?,?)");

preparedStatement.setString(1, user.getUsername());

preparedStatement.setString(2, user.getPassword());

preparedStatement.setString(3, user.getEmail());

preparedStatement.setString(4, user.getNameSurname());

preparedStatement.setString(5, user.getAge() + "");

preparedStatement.setString(6, user.getGender());

- We use Databaseconnector class on java in order to connect database. Codes are given below

public class DatabaseConnector {

public static final long serialVersionUID = -997894563384L;

public static String userName = "pdinfo\_remote";

public static String password = "cs353";

public synchronized static Connection getConnection() {

try {

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection(

"jdbc:mysql://213.159.1.91:3306/pdinfo\_database\_project?characterEncoding=UTF-8&useSSL=false&?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=Turkey",

userName, password);

return con;

} catch (ClassNotFoundException | SQLException ex) {

ex.printStackTrace();

return null;

}

}

public synchronized static void close(Connection con) {

try {

con.close();

} catch (Exception ex) {

}

}

}

- In our project, we use defaultGUI. Here is one of the page’s GUI.

<ui:composition xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:f="http://java.sun.com/jsf/core"

xmlns:ui="http://java.sun.com/jsf/facelets"

xmlns:p="http://primefaces.org/ui"

template="/WEB-INF/template.xhtml">

<ui:define name="title">Questions</ui:define>

<ui:define name="content">

<h:form id="form">

<div class="ui-g" style="margin-left: 15%; margin-right: 15%; margin-top: 10px; ">

<div class="ui-g-12 ui-md-12">

<h2 style="text-align: center;"> Questions</h2>

</div>

<div class="ui-g-12 ui-md-3" style="background-color: #e5e5e5;">

<p:dataTable id="trending" emptyMessage="Empty" var="question" value="#{questionsBean.trendingQuestions}" selectionMode="single"

selection="#{questionsBean.tempSelectedQuestion}" rowKey="#{question.rowKey}"

scrollable="true" scrollHeight="250" style="margin-bottom: 3%;"

rowIndexVar="rowIndex">

<f:facet name="header">

Trending Questions

</f:facet>

<p:column headerText="No" style="width: 15%; text-align: center">

#{rowIndex+1}

</p:column>

<p:column headerText="Title" style="text-align: center">

<h:outputText value="#{question.title}" />

</p:column>

</p:dataTable>

<p:dataTable id="recent" emptyMessage="Recent questions list is empty" var="question" value="#{questionsBean.recentQuestions}" selectionMode="single"

selection="#{questionsBean.tempSelectedQuestion}" rowKey="#{question.rowKey}"

scrollable="true" scrollHeight="250" style="margin-bottom: 1%;"

rowIndexVar="rowIndex">

<f:facet name="header">

Recent Questions

</f:facet>

<p:ajax event="rowSelect" listener="#{questionsBean.showMore(question)}" global="false" />

<p:column headerText="No" style="width: 15%; text-align: center">

#{rowIndex+1}

</p:column>

<p:column headerText="Title" style="text-align: center">

<h:outputText value="#{question.title}" />

</p:column>

</p:dataTable>

</div>

<div class="ui-g-12 ui-md-9" style="background-color: #f2f2f2">

<p:panel id="content" style="margin: 0px; border: none;">

<p:panel rendered="#{questionsBean.selectedQuestion != null}" >

<h:outputText value="#{questionsBean.selectedQuestion.title}" />

<p:separator style="opacity: 0"/>

<h:outputText value="#{questionsBean.selectedQuestion.questionArea}" />

<p:panelGrid columns="3" columnClasses="ui-grid-col-7, ui-grid-col-3, ui-grid-col-2" layout="grid" styleClass="ui-panelgrid-blank"

style="border:0px none; background-color:transparent; margin: 0%; margin-bottom: -20px;">

<h:outputText value="#{questionsBean.selectedQuestion.user.nameSurname}" style="color: #00838f"/>

<p:spacer/>

<p:commandButton value="Close" actionListener="#{questionsBean.closeQuestion()}" update="template\_form:form:comments" style="width: 100%;"/>

</p:panelGrid>

<p:panelGrid columns="1" columnClasses="ui-grid-col-12" layout="grid" styleClass="ui-panelgrid-blank"

style="border:0px none; background-color:transparent; margin: 0%; margin-bottom: -20px;">

<p:inputTextarea rows="5" value="#{questionsBean.commentText}" queryDelay="750" minQueryLength="4"

style="width: 100%; background-color: #f5f5f5; margin-top: 30px;"/>

<p:commandButton value="Send" styleClass="secondary" actionListener="#{questionsBean.sendComment()}" style="width: 100%; float: right"/>

</p:panelGrid>

<p:dataTable id="comments" emptyMessage="Empty" var="comment" value="#{questionsBean.selectedQuestion.comments}"

rowKey="#{comment.index}"

rows="10" paginator="true" rowsPerPageTemplate="10, 50, 100, 250" paginatorPosition="top"

paginatorTemplate="{CurrentPageReport} {PreviousPageLink} {PageLinks} {NextPageLink} {RowsPerPageDropdown}"

scrollable="true" scrollHeight="640" style="font-size:12px; margin-top: 50px;"

rowIndexVar="rowIndex">

<f:facet name="header">

Comments

</f:facet>

<p:column headerText="" style="text-align: left">

<p:panelGrid columns="1" columnClasses="ui-grid-col-12" layout="grid" styleClass="ui-panelgrid-blank"

style="border:0px none; background-color:transparent; margin: 1%">

<h:outputText value="#{comment.commentArea}" />

</p:panelGrid>

</p:column>

</p:dataTable>

</p:panel>

<p:dataTable id="questionsTable" rendered="#{questionsBean.selectedQuestion == null}" emptyMessage="Empty" var="question" value="#{questionsBean.recentQuestions}"

rowKey="#{question.rowKey}"

rows="10" paginator="true" rowsPerPageTemplate="10, 50, 100, 250" paginatorPosition="top"

paginatorTemplate="{CurrentPageReport} {PreviousPageLink} {PageLinks} {NextPageLink} {RowsPerPageDropdown}"

scrollable="true" scrollHeight="640" style="font-size:12px;"

rowIndexVar="rowIndex">

<f:facet name="header">

Questions

</f:facet>

<p:column headerText="" style="text-align: left">

<p:panelGrid columns="1" columnClasses="ui-grid-col-12" layout="grid" styleClass="ui-panelgrid-blank"

style="border:0px none; background-color:transparent; margin: 1%">

<h:outputText value="#{question.title}" />

<h:outputText value="#{questionsBean.questionSummary(question.questionArea)}" />

<p:panelGrid columns="3" columnClasses="ui-grid-col-7, ui-grid-col-3, ui-grid-col-2" layout="grid" styleClass="ui-panelgrid-blank"

style="border:0px none; background-color:transparent; margin: 0%; margin-bottom: -20px;">

<h:outputText value="#{question.user.nameSurname}" style="color: #00838f"/>

<p:commandButton value="#{userBean.user != null and userBean.user.followList.contains(question.user) ? 'Unfollow User':'Follow User'}"

actionListener="#{questionsBean.followUser(question.user)}" style="width: 100%;"/>

<p:commandButton value="Read More" actionListener="#{questionsBean.showMore(question)}" style="width: 100%;"/>

</p:panelGrid>

</p:panelGrid>

</p:column>

</p:dataTable>

</p:panel>

</div>

</div>

<h:form>

<p:dialog id="regOrLogDialog" style="min-width: 400px; max-width: 500px" header="Login / Register" widgetVar="regOrLogDialog"

modal="true" showEffect="fade" dynamic = "true" hideEffect="fade" resizable="false">

<p:panelGrid columns="1" columnClasses=" ui-grid-col-12"

layout="grid" styleClass="ui-panelgrid-blank" style="margin: 2%; border:0px none; background-color:transparent;">

<p:outputLabel value="You need to register/login to follow" />

</p:panelGrid>

<p:separator style="opacity: 0.4;"/>

<p:panelGrid columns="3" columnClasses=" ui-grid-col-3, ui-grid-col-5, ui-grid-col-4"

layout="grid" styleClass="ui-panelgrid-blank" style="border:0px none; background-color:transparent;">

<p:spacer/>

<p:commandButton value="register/login" style="min-width: 100px" action="login\_register.xhtml"

onclick="PF('regOrLogDialog').hide();"/>

<p:commandButton value="Close" style="min-width: 100px" onclick="PF('regOrLogDialog').hide();" />

</p:panelGrid>

</p:dialog>

</h:form>

</h:form>

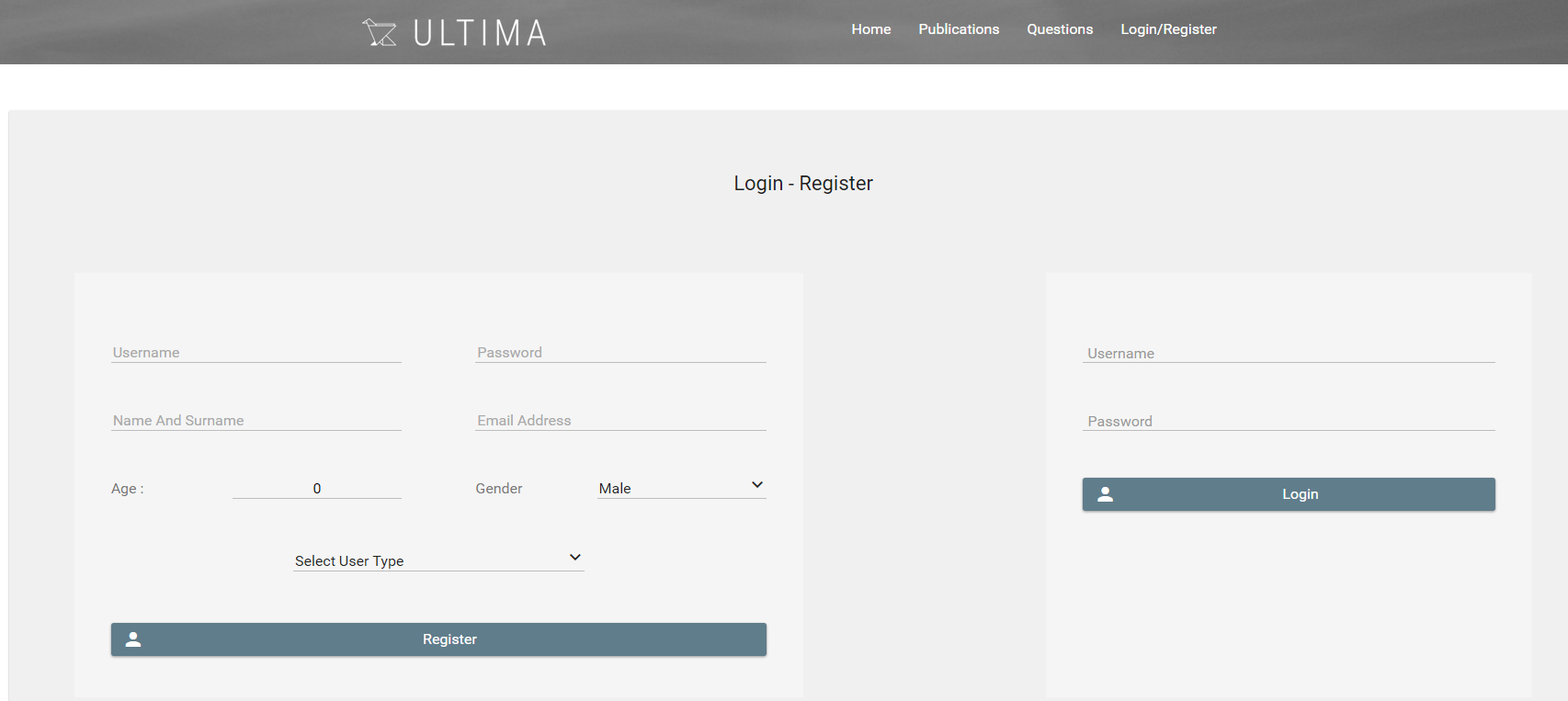
</ui:define>

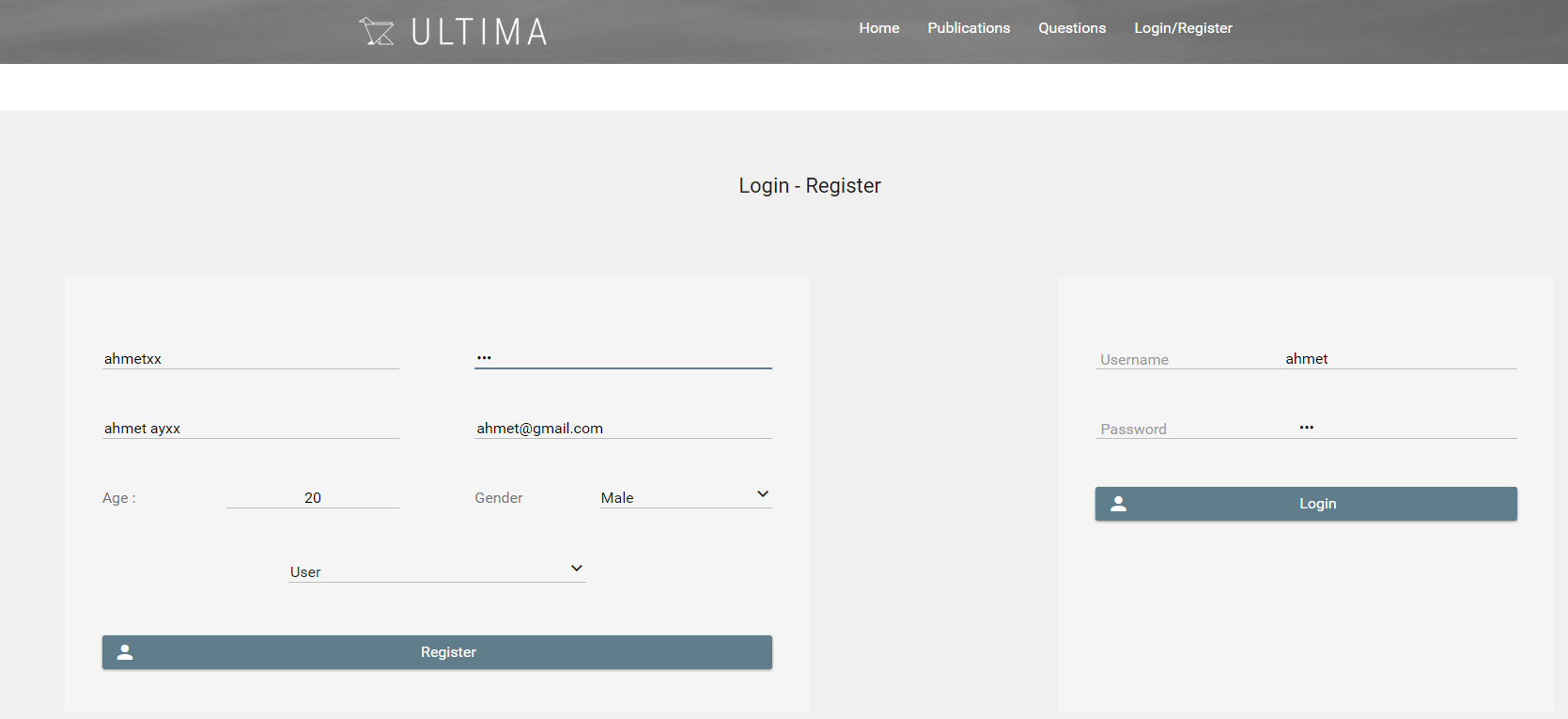
</ui:composition>

**USER MANUAL AND SAMPLE OUTPUT**

**Login Page**

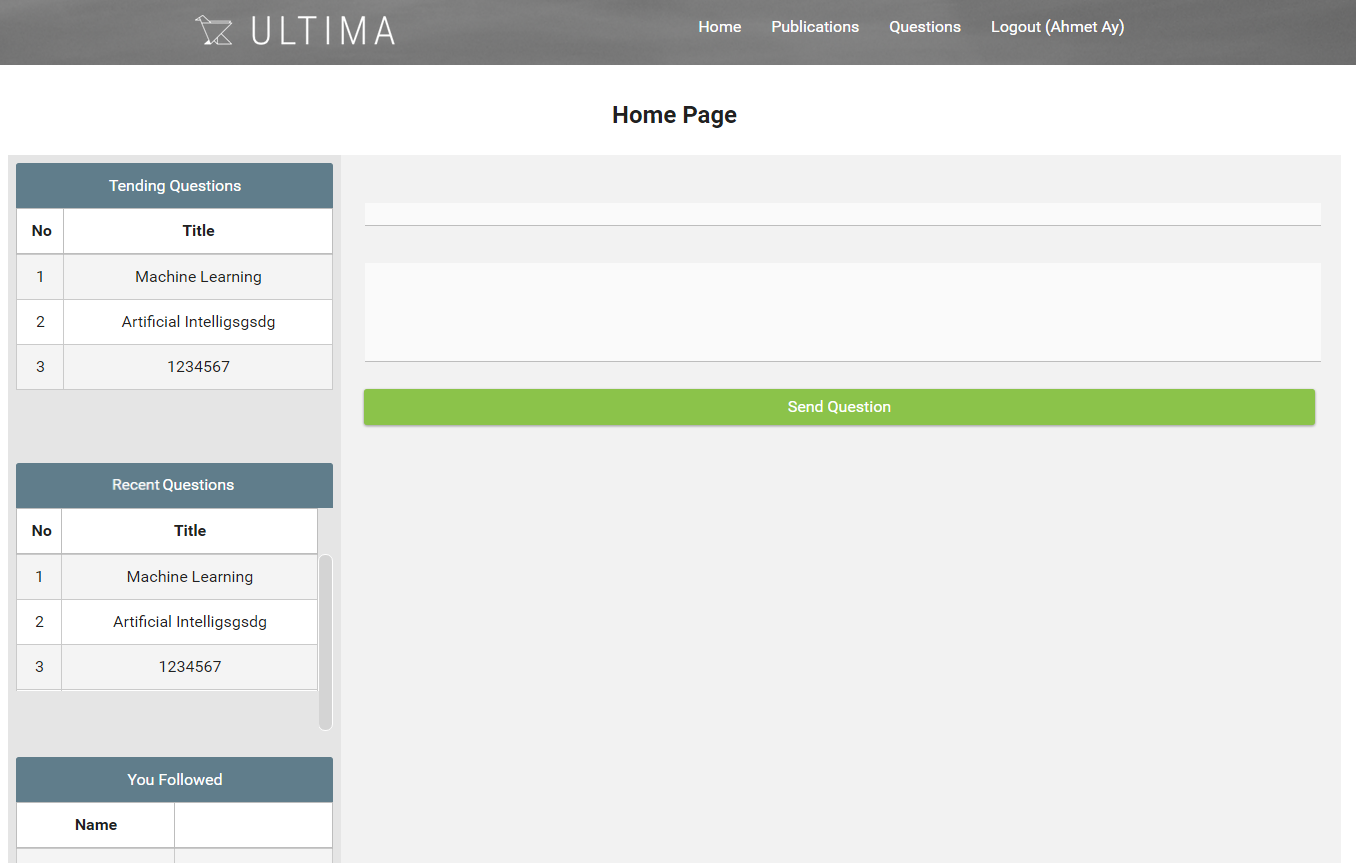
It is the process by which an individual gains access to a system by identifying and authenticating themselves. User can signup by entering username, password, name-surname, e-mail adress, gender and user type (user or author) and user credentials are typically some form of "username" and a matching "password" to login the website.





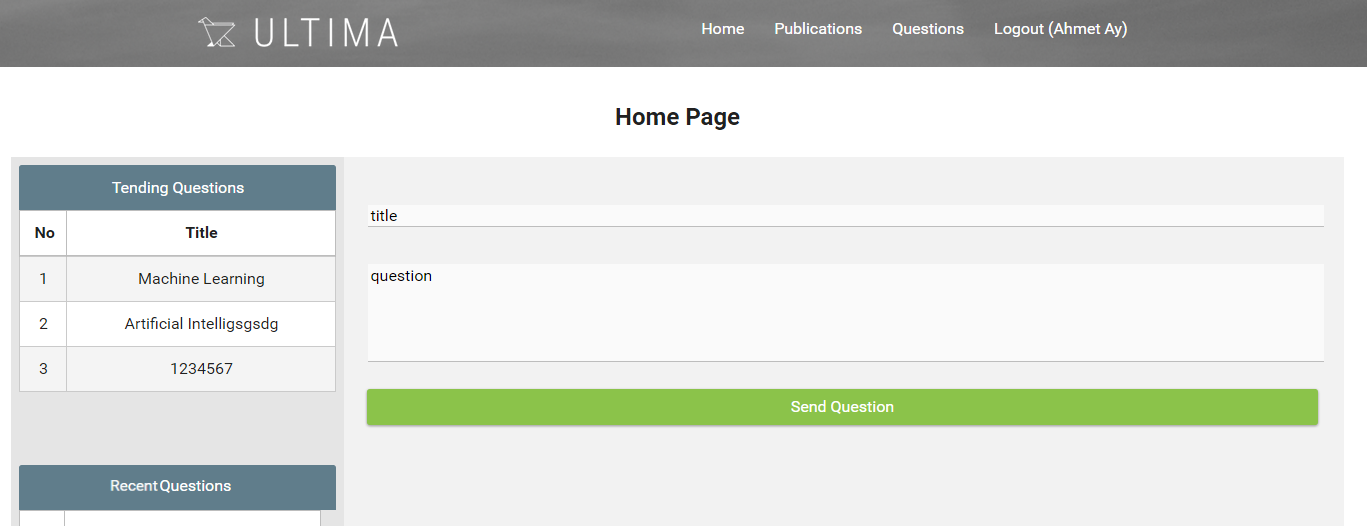
**Home Page**

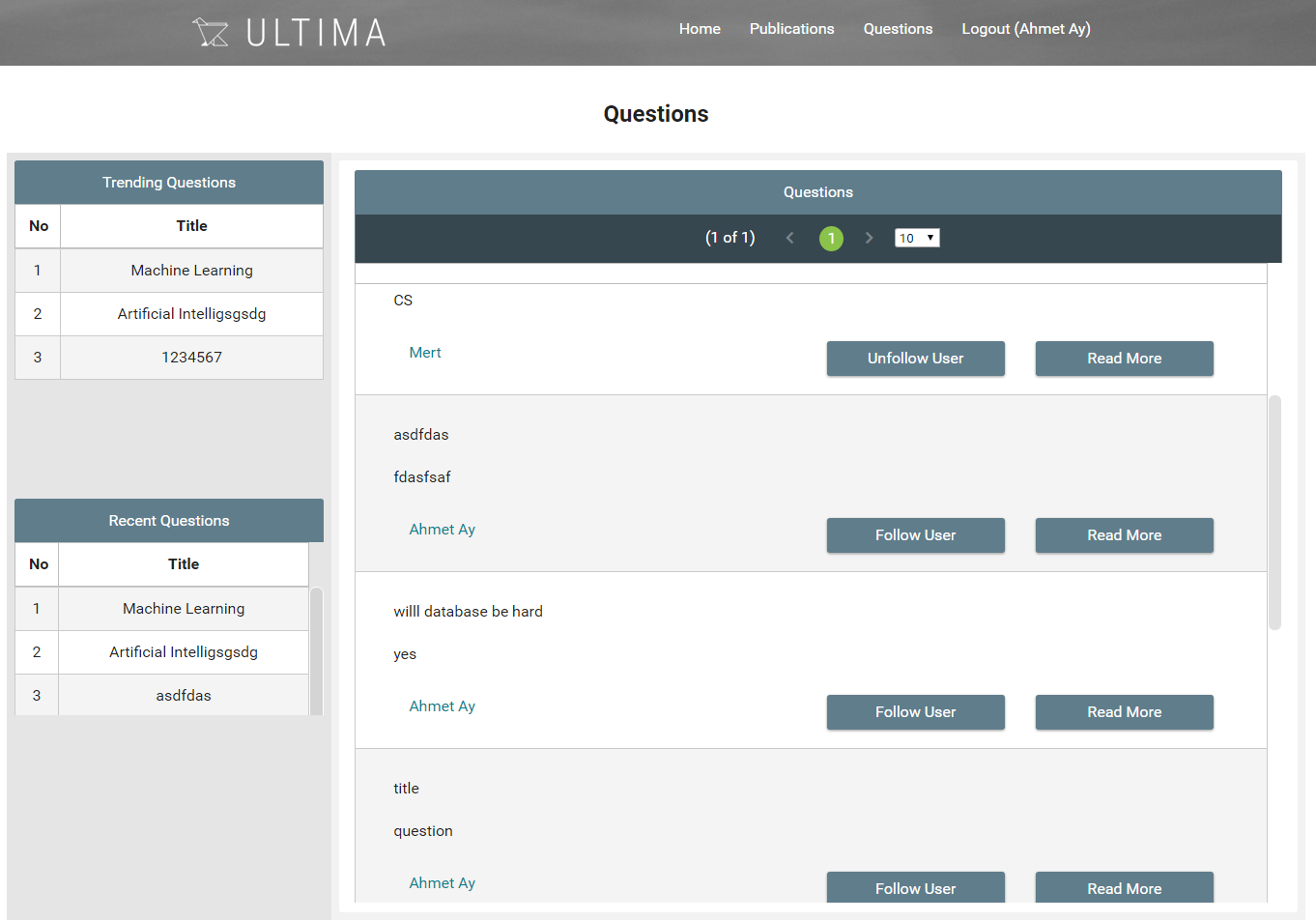
A home page is the main page a visitor navigating to a website. The home page is used to facilitate navigation to other pages on the site by providing links to prioritized and recent articles and pages. This website presents some of features. In the left side, Trending Questions, Recent Questions and following users are shown. Trending questions calculated by counting most pressed question. Recent question is calculated by determining last entered question date. In the middle of the page, there is a box that user can send new question. User can define question headline and its detail description. After user write them, he can click on the send button and question will be uploaded to the app. Also, in the top of the app, there are four option to be choosed: Home, Publications, Questions and Logout. By clicking on these buttons, user can go to these pages.



**Questions**

Sending Question

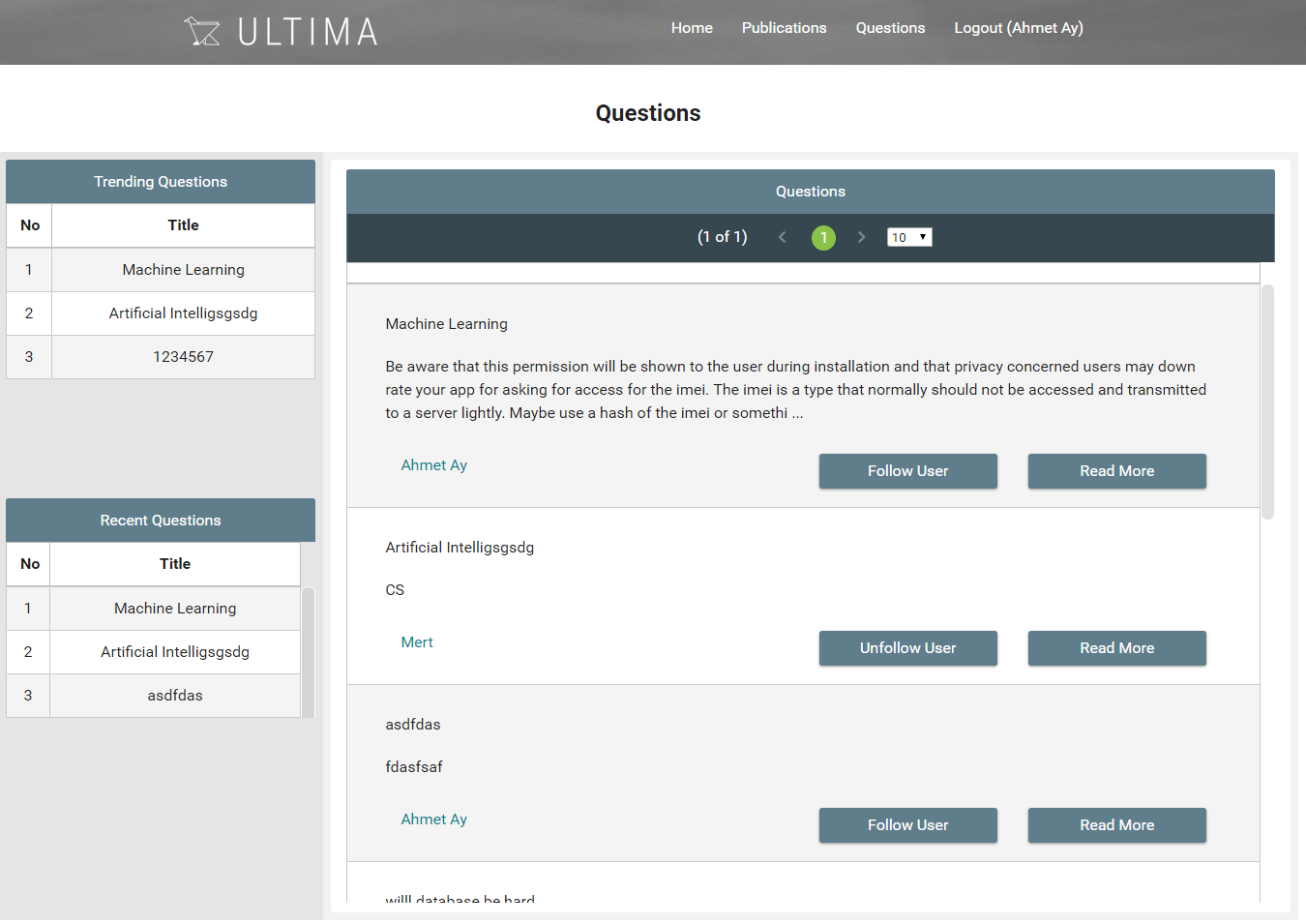


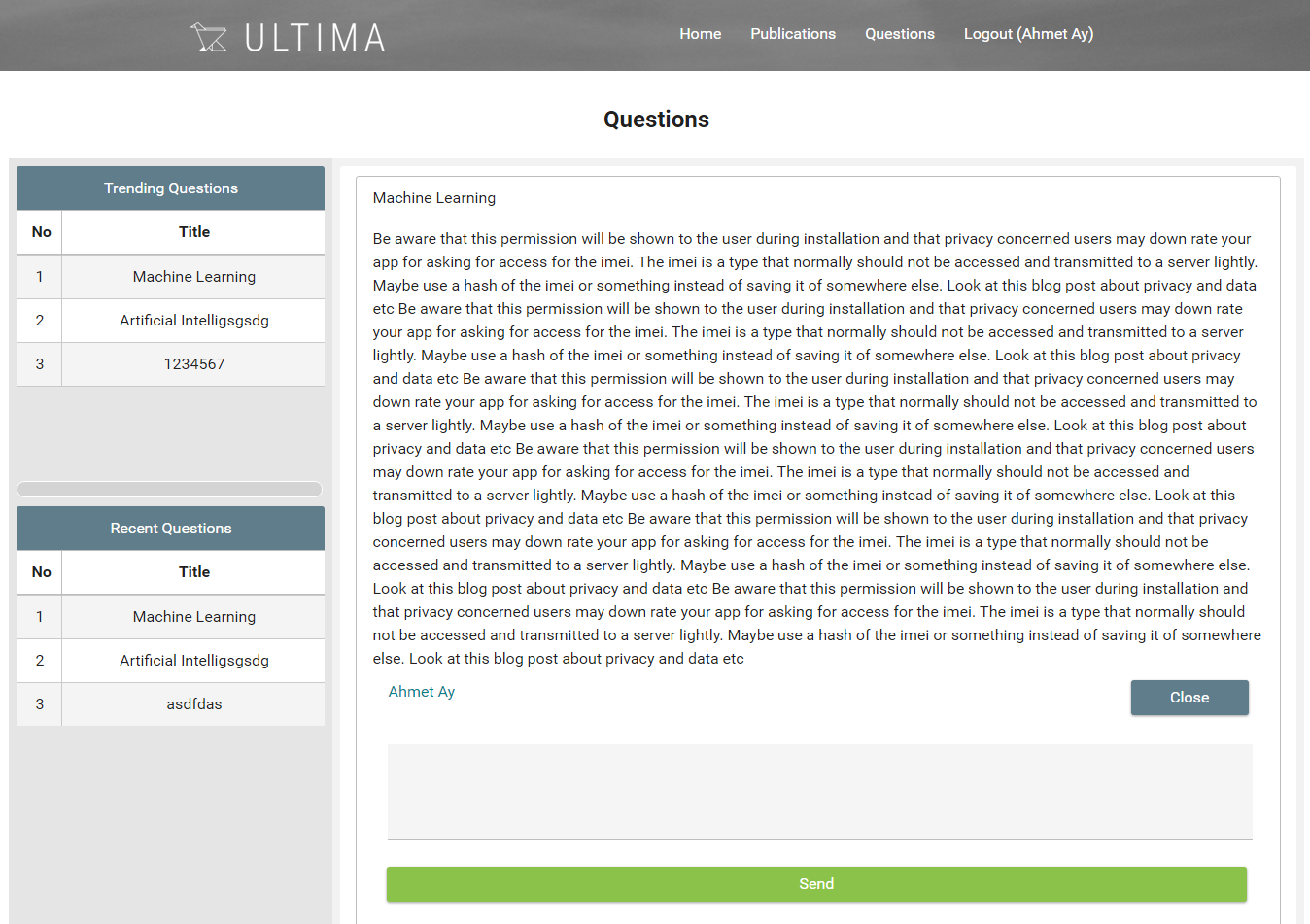


-Question is displayed at the last box in the Question. For each question, user can follow/unfollow the other user that sent the question by clicking on the button. All questions are listed in the middle of the page and user can go down by using scroll bar. User also can read more detailed the question by clicking on read more button.

Read question more

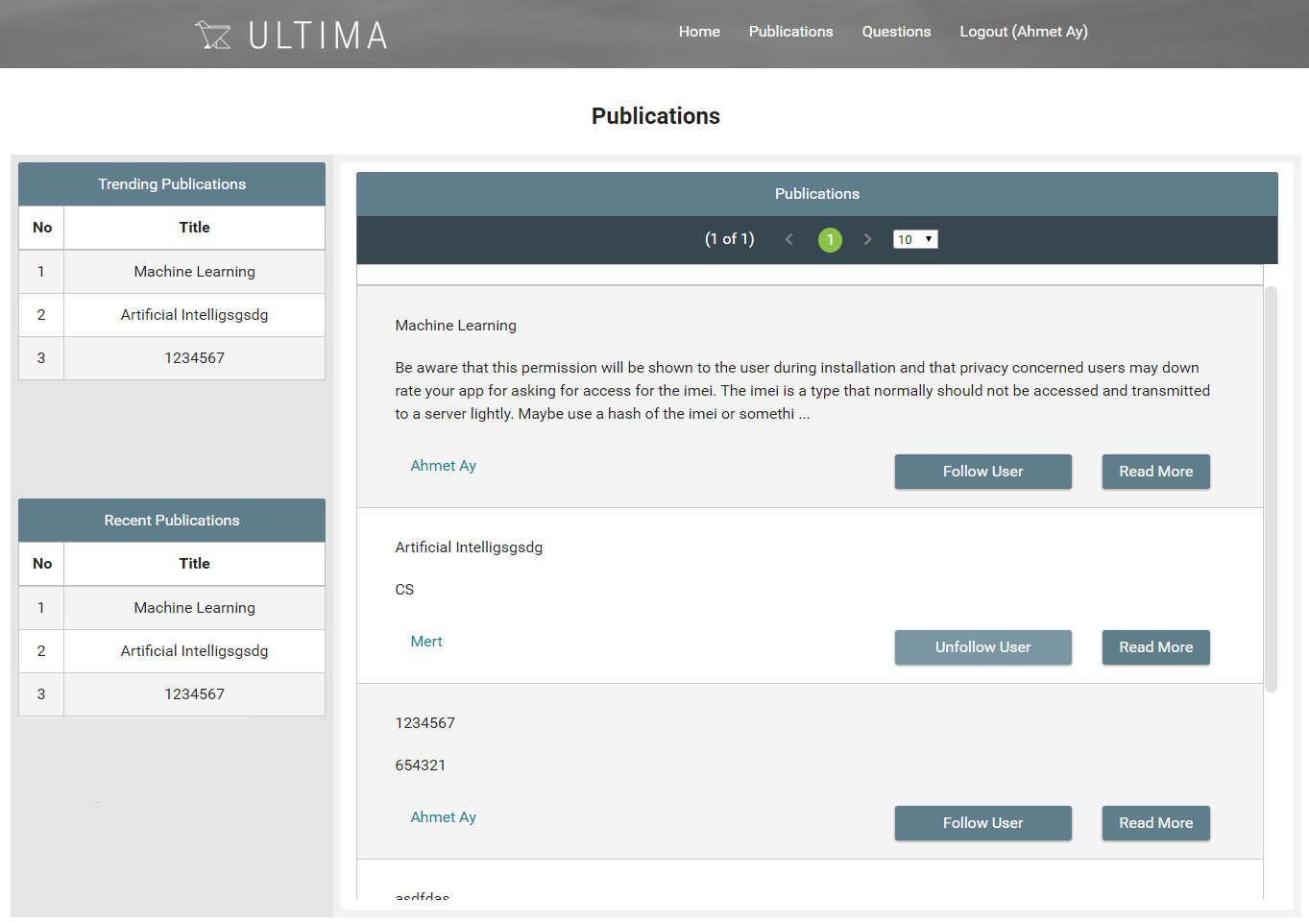
After clicking on the read more button, question detail is shown in the same frame. User can read long description, headline and can go back to the question page by back button. Recent Questions and Trending Questions are also still shown in the left side of the page. In the bottom of the page, there is a box for the comment purpose and its send button.

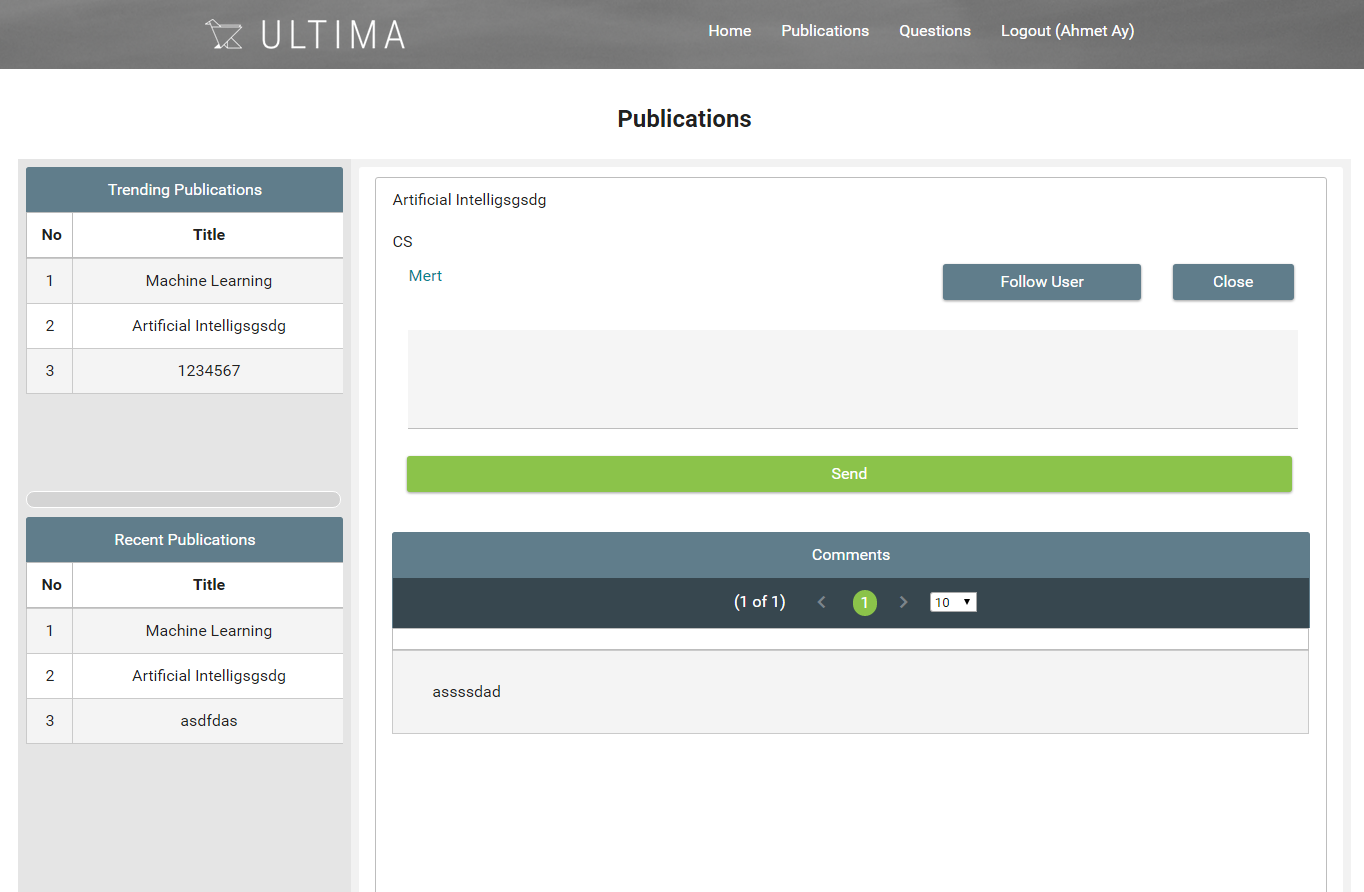




**Publications**

It is similar as Question part. It is displayed at the last box in the Publications. For each publication, user can follow/unfollow the other user that sent the question by clicking on the button. All publications are listed in the middle of the page and user can go down by using scroll bar. User also can read more detailed the publication by clicking on read more button.

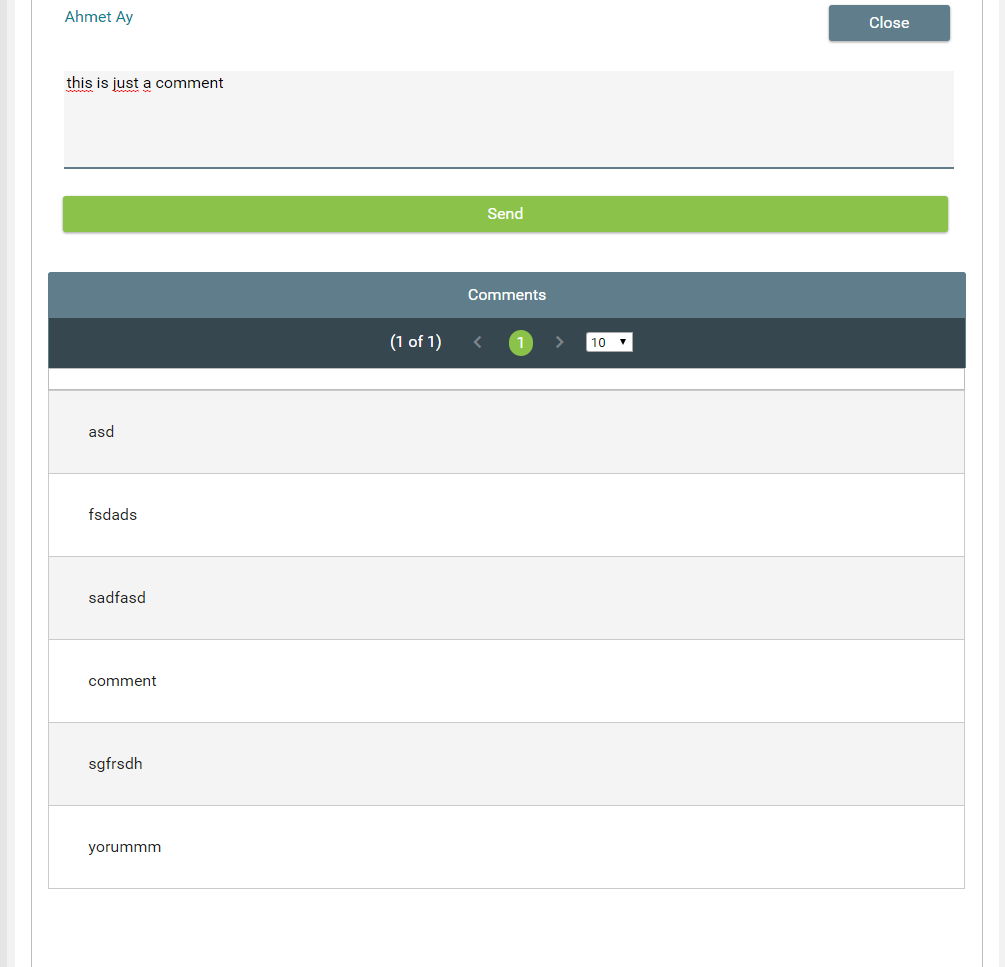


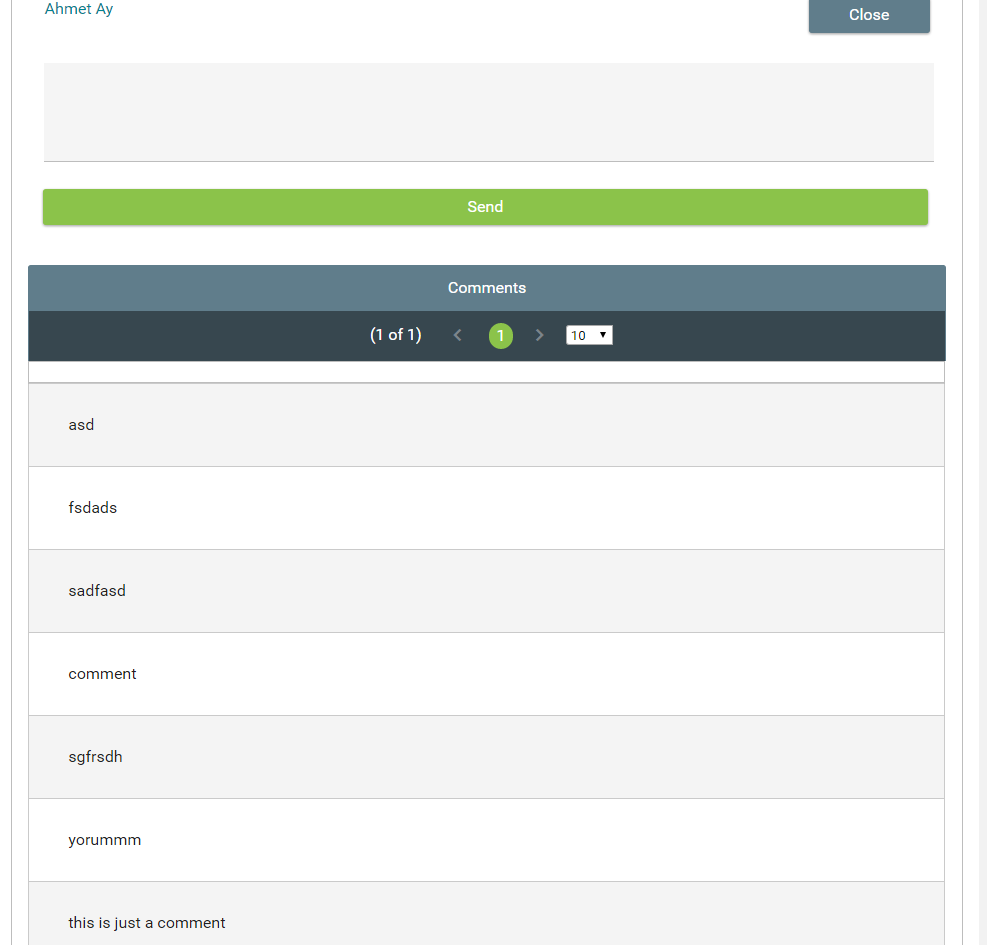


After clicking on the read more button, question detail is shown in the same frame. User can read long description, headline and can go back to the question page by back button. Recent Publications and Trending Publications are also still shown in the left side of the page. In the bottom of the page, there is a box for the comment purpose and its send button.

**Comment**

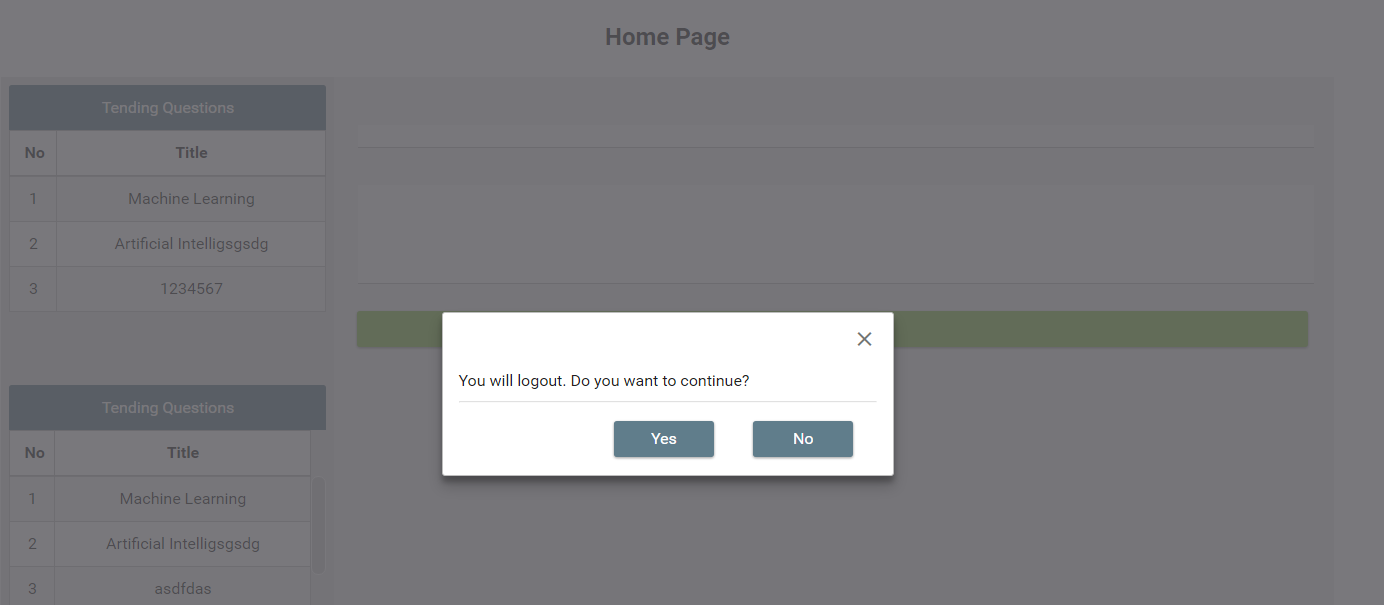
Comment box is appeared to chosen question. Registered user writes what she/he deserves and by clicking send button question is sent. Sent comment is appeared under the chosen question according to oldest to newest. Number of comments that is shown on each page can be choose by user.





-comment is added.

**Log Out warning message**

In order to log out system, User should click the logout button on top right of the site. To prevent missclicking , Pop up appears to just be sure user really wants to login

*Finished*