Gemini Team Project Plan: Employee Recognition

Team Members:
Joel Smith
Bryan Nguyen

The Gemini Team will be creating a web application for creating and managing employee recognition awards. Our team members have some experience with web development and databases, and this project will allow us to further develop those skills by applying them towards building a website with extensive functionality, including user authentication and administration, execution of third-party applications, email sending, and database query operations.

Users of this application will have accounts that store their name and an image of their signature. Once logged in, a user will be able to create personalized awards for employees by selecting a type of award and filling in information about the recipient. When the information is submitted, a customized PDF certificate will be generated using the LaTeX typesetting system and emailed to the recipient. Sent awards will be recorded in a database, and users will be able to view and delete awards they have sent. Additionally, an administrator class of users will be able to manage user accounts, perform queries on the database of sent awards, and display these queries as reports with graphical charts or export them as CSV files.

The user portion of the website will consist of the following pages:

- A user login page, allowing login with an email address and password
- A password recovery page, where a user can request a password reset link to be emailed to them
- An awards page, where a user can view sent awards and delete them
- A user settings page, where a user can change their name or password
- An award creation page, where a user can input information (selecting an award type and inputting the recipient’s name and email address) to create and send an award

The admin portion of the website will consist of the following pages:

- An admin login page, allowing login with an email address and password
- A password recovery page, where an administrator can request a password reset link to be emailed to them
- A user management page, where users can be added, edited, and deleted
- An admin management page, where administrators can be added, edited, and deleted
- A query page, where data on the awards that have been sent can be requested
- A results page, where query results are displayed, including graphical charts, and can be exported as a CSV file

The backing database will include the following tables and attributes:

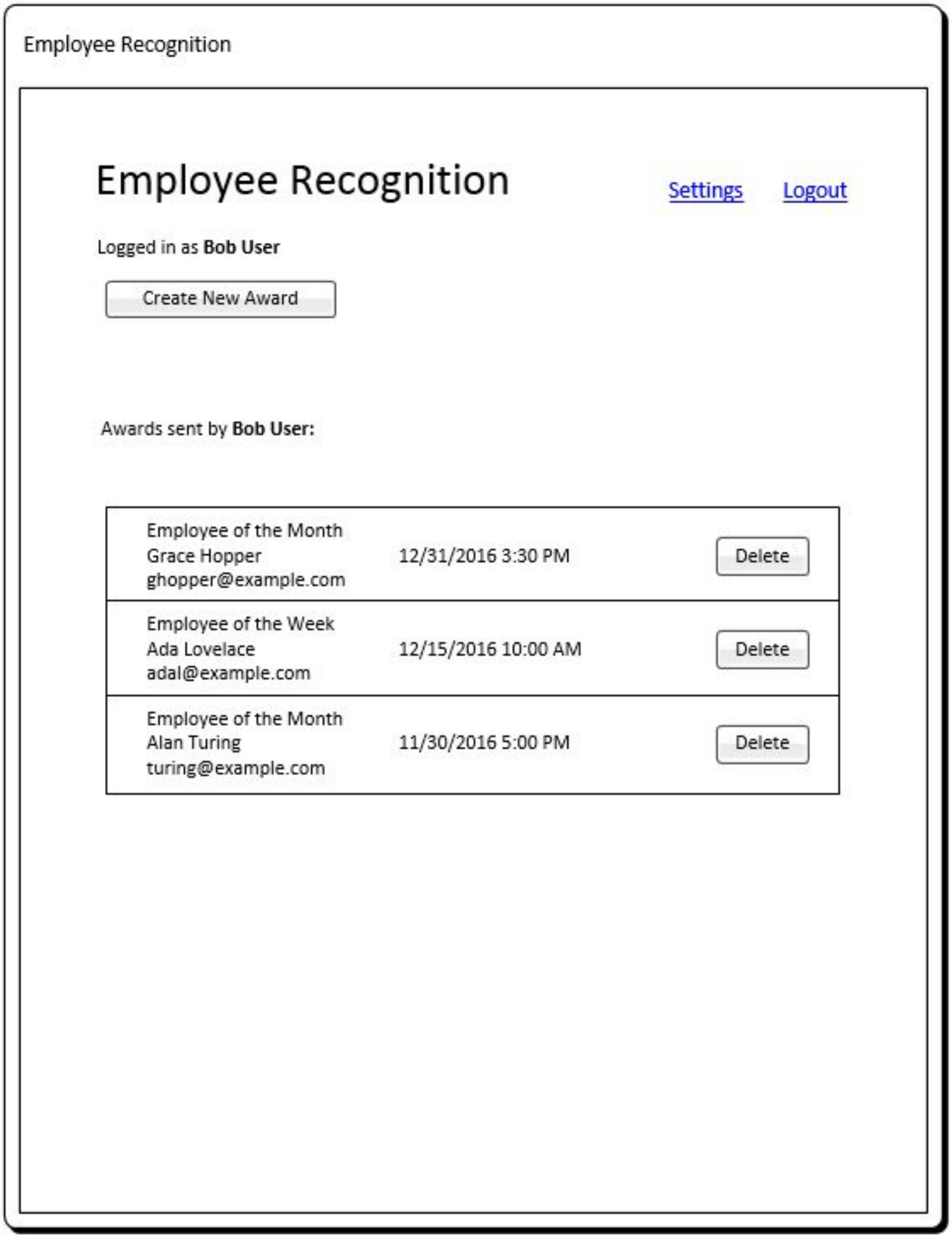
- User accounts
    - Email address
    - Password hash
    - Name
    - Account creation time
    - Signature
- Administrator accounts
    - Email address
    - Password hash
    - Account creation time
- Award classes
- Award entries
    - Award type
    - Recipient name
    - Recipient email address
    - Creating user
    - Time granted

The application back-end will be built in Node.js. The website front-end will be written in HTML, CSS, JavaScript, and jQuery. The backing database will be implemented using SQLite. The Google Charts API will be used for displaying charts on query reports.

Tasks will be delegated to the team members according to the following schedule:

Week	Joel’s tasks	Time (h)	Bryan’s tasks	Time (h)
3	Set up and configure Node.js server. Set up SQLite for use with Node.js. Record progress report video.	12	Design database schema. Set up user and admin authentication systems.	12
4	Design, create, and test admin interface pages for managing users.	12	Design, create, and test user interface pages for creating and sending awards. Record progress report video.	12
5	Implement sending emails with file attachments from Node.js. Record progress report video.	12	Implement award certificate generation: design LaTeX templates and implement PDF creation from user input.	15
6	Prepare test data set for use with querying functionality. Begin implementing querying functionality.	12	Integration testing for award generation and emailing. Begin implementing querying functionality. Write documentation for mid-project report.	15
7	Finish implementing querying functionality. Begin implementing Google Charts graphics.	15	Finish implementing querying functionality. Record progress report video.	12
8	Finish implementing Google Charts graphics. Record progress report video.	15	Implement exporting results in CSV format.	12
9	Final testing, bug fixes, and graphical/user interface improvements.	12	Final testing, bug fixes, and graphical/user interface improvements. Record progress report video.	12
10	Prepare final report and final submission.	10	Prepare final report and final submission.	10
	Total:	100	Total:	100

Below is a mockup of the page the user sees after logging in, which displays the awards already sent.



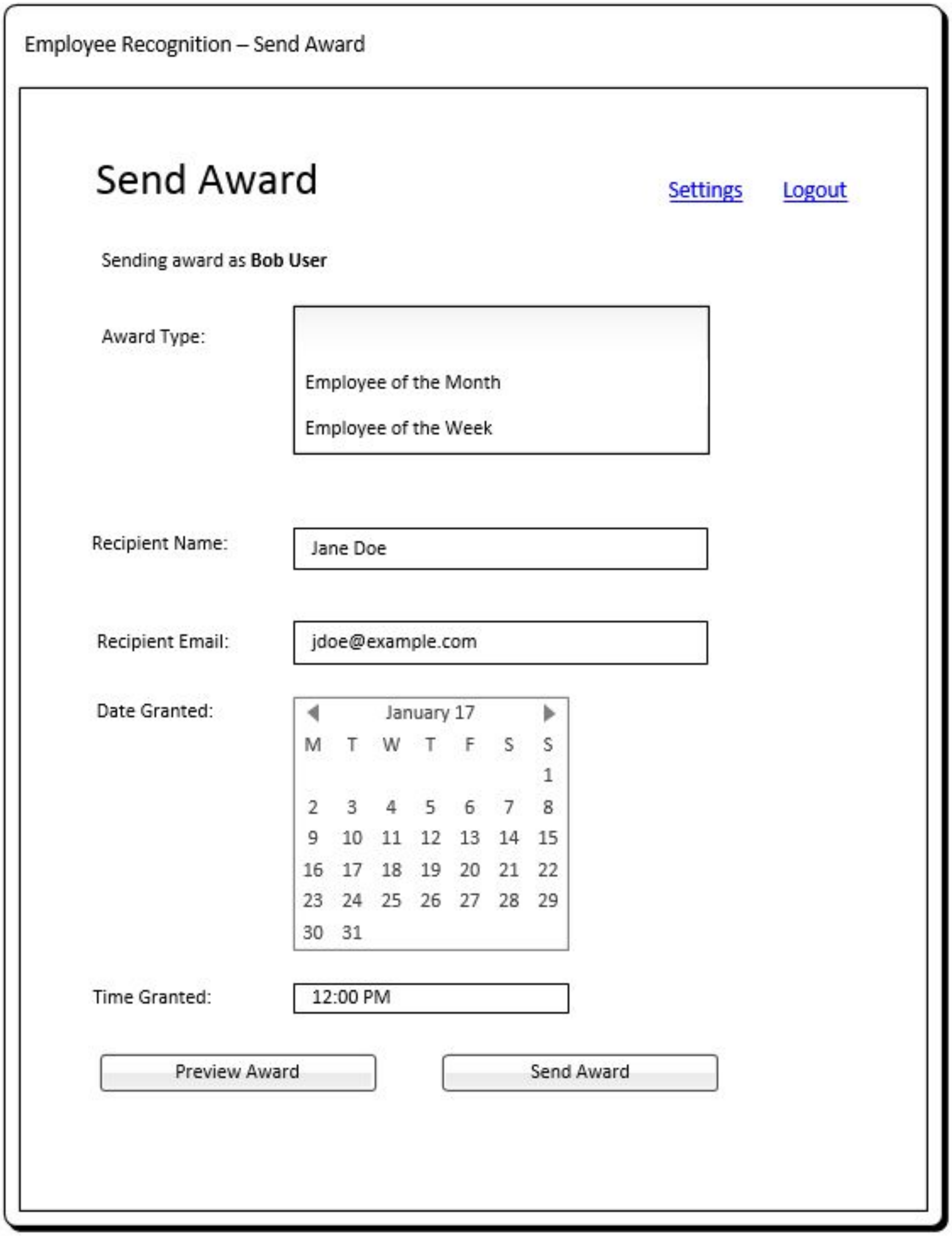
Employee Recognition

Logged in as **Bob User**

Awards sent by **Bob User:**

Employee of the Month Grace Hopper ghopper@example.com	12/31/2016 3:30 PM	Delete
Employee of the Week Ada Lovelace adal@example.com	12/15/2016 10:00 AM	Delete
Employee of the Month Alan Turing turing@example.com	11/30/2016 5:00 PM	Delete

Below is a mockup of the page from which a user can send an award.



Employee Recognition – Send Award

Send Award

Sending award as **Bob User**

Award Type: Employee of the Month

Employee of the Week

Recipient Name: Jane Doe

Recipient Email: jdoe@example.com

Date Granted: January 17

M	T	W	T	F	S	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Time Granted: 12:00 PM

By following the plan outlined in this document, we expect to produce a high-quality web application for sending and tracking employee recognition awards and for making queries about awards that have been sent. We estimate that this project can be completed within 200 person-hours.