Project Management System

Arpit Vaghela
Ahmedabad University
arpitsinh.v@ahduni.edu.in
AU1841034

Kaushal Patil
Ahmedabad University
kaushal.p@ahduni.edu.in
AU1841040

Dhruvil Dave
Ahmedabad University
dhruvil.d@ahduni.edu.in
AU1841003

***Abstract*—This is a project management system written using React.js for frontend, Flask and GraphQL for backend and PostgreSQL as database. This project was made as a part of subjects Database Management Systems (CSE250) and Database Management Systems Lab (CSE251).**

***Index Terms*—component, formatting, style, styling, insert**

I. Table Design

TABLE I
USERS

Column	Type	Nullable	Default
username	text	not null	
firstname	text	not null	
lastname	text	not null	
password	text	not null	
emailid	text	not null	
profilepic	byte array		

1) Indexes:
   a) PRIMARY KEY (username)
   b) UNIQUE CONSTRAINT (emailid)
2) Check Constraints:
   a) emailid
   b) firstname
   c) lastname
   d) username
3) Referenced by:
   a) assignedto VI
   b) board VIII
   c) member III
   d) note IX
   e) project II
4) Triggers:
   a) add_board AFTER INSERT ON users FOR EACH ROW EXECUTE FUNCTION add_board()
   b) create_hash BEFORE INSERT OR UPDATE ON users FOR EACH ROW EXECUTE FUNCTION create_hash()

Fig. 1. Example of a figure caption.

TABLE II
PROJECT

Column	Type	Nullable	Default
projectid	integer	not null	serial
name	text	not null	
shortdescription	text		
longdescription	text		
createdon	date		
path	text		
createdby	text		
status	enum(project_status)	not null	completed

1) Indexes:
   a) PRIMARY KEY (projectid)
   b) UNIQUE CONSTRAINT (name, createdby)
2) Foreign Key Contraints:
   a) createdby REFERENCES users(username)
3) Check Constraints:
   a) emailid
   b) firstname
   c) lastname
   d) username
4) Referenced by:
   a) board VIII
   b) member III
   c) projectfiles IV
5) Triggers:
   a) add_board AFTER INSERT ON project FOR EACH ROW EXECUTE FUNCTION add_board()
   b) add_leader AFTER INSERT ON project FOR EACH ROW EXECUTE FUNCTION add_leader()

TABLE III
MEMBER

Column	Type	Nullable	Default
username	text	not null	
projectid	integer	not null	
role	enum(role_type)		

1) Indexes:
   a) PRIMARY KEY (username, projectid)
2) Foreign Key Contraints:
   a) projectid REFERENCES users(projectid)
   b) username REFERENCES users(username)
3) Referenced by:
   a) task V
4) Triggers:
   a) add_board AFTER INSERT ON project FOR EACH ROW EXECUTE FUNCTION add_board()
   b) add_leader AFTER INSERT ON project FOR EACH ROW EXECUTE FUNCTION add_leader()

TABLE IV
PROJECTFILES

Column	Type	Nullable	Default
fileid	integer	not null	serial
filename	text	not null	
file	byte array		
lastupdated	timestamp without time zone	not null	serial
projectid			

1) Indexes:
   a) PRIMARY KEY (fileid)
2) Foreign Key Contraints:
   a) projectid REFERENCES project(projectid)
3) Check Constraints:
   a) filename
4) Triggers:
   a) update_lastupdated_files BEFORE INSERT OR UPDATE ON projectfiles FOR EACH ROW EXECUTE FUNCTION update_lastupdated_files()

TABLE V
TASK

Column	Type	Nullable	Default
taskid	integer	not null	serial
title	text	not null	
description	text		
starttime	timestamp without time zone		now()
endtime	timestamp without time zone		
status	enum(status_type)		active
completiontime	timestamp without time zone		
priority	enum(priority_type)		normal
assignedby	text	not null	
projectid	integer	not null	

1) Indexes:
   a) PRIMARY KEY (taskid)
   b) UNIQUE CONSTRAINT (title, assignedby, project)
2) Foreign Key Contraints:
   a) (assignedby, projectid) REFERENCES member(username, projectid)
3) Check Constraints:
   a) starttime
4) Referenced by:
   a) board VI
   b) preqtask VII
5) Triggers:
   a) add_task BEFORE INSERT OR UPDATE ON task FOR EACH ROW EXECUTE FUNCTION add_task()
   b) chech_projectstatus AFTER INSERT OR UPDATE ON task FOR EACH ROW EXECUTE FUNCTION check_projectstatus()
   c) update_status AFTER UPDATE ON task FOR EACH ROW EXECUTE FUNCTION update_status()

TABLE VI
ASSIGNEDTO

Column	Type	Nullable	Default
taskid	integer	not null	
username	text	not null	

1) Indexes:
   a) PRIMARY KEY (taskid, username)
2) Foreign Key Contraints:
   a) taskid REFERENCES task(projectid)
   b) username REFERENCES users(username)

TABLE VII
PREQTASK

Column	Type	Nullable	Default
task	integer	not null	
preqtask	integer	not null	

1) Indexes:
   a) PRIMARY KEY (task, preqtask)
2) Foreign Key Contraints:
   a) task REFERENCES task(taskid)
   b) preqtask REFERENCES task(taskid)
3) Triggers:
   a) task_status_to_inactive AFTER INSERT OR UPDATE ON preqtask FOR EACH ROW EXECUTE FUNCTION change_statuson_preqtask()

TABLE VIII
BOARD

Column	Type	Nullable	Default
boardid	integer	not null	serial
title	text	not null	
description	text		
username	text		
projectid	integer		

1) Indexes:
   a) PRIMARY KEY (boardid)
2) Foreign Key Contraints:
   a) projectid REFERENCES project(projectid)
   b) username REFERENCES users(username)
3) Referenced by:
   a) note IX
4) Triggers:
   a) check_board BEFORE INSERT ON board FOR EACH ROW EXECUTE FUNCTION check_board()

TABLE IX
BOARD

Column	Type	Nullable	Default
noteid	integer	not null	serial
title	text	not null	
description	text		
color	text		
createdby	text	not null	
boardid	integer		

1) Indexes:
   a) PRIMARY KEY (noteid)
   b) UNIQUE CONSTRAINT (title, description)
2) Foreign Key Contraints:
   a) boardid REFERENCES board(boardid)
   b) createdby REFERENCES users(username)