Travel Tracker

Olivia Folsom, Tasmia Iqbal, Panhapich Leang, Olivia Tarsillo

Deliverable Version 3.0

Chapter 1

Purpose

Travel Tracker is a web application that aims to streamline vacation planning into one central location. Our application allows users to manage the logistics of their trip(s), such as budgeting, flight information, stay information, itinerary, and track interests such as excursions, activities, and additional notes. Travel Tracker is a user-friendly planner that eases travel stress and encourages strong organization. Our budgeting services, equipped with real-time currency conversions, advise users when to save money on international trips and permit users to set allowances, alleviate uncertainty, and allow users to budget accordingly.

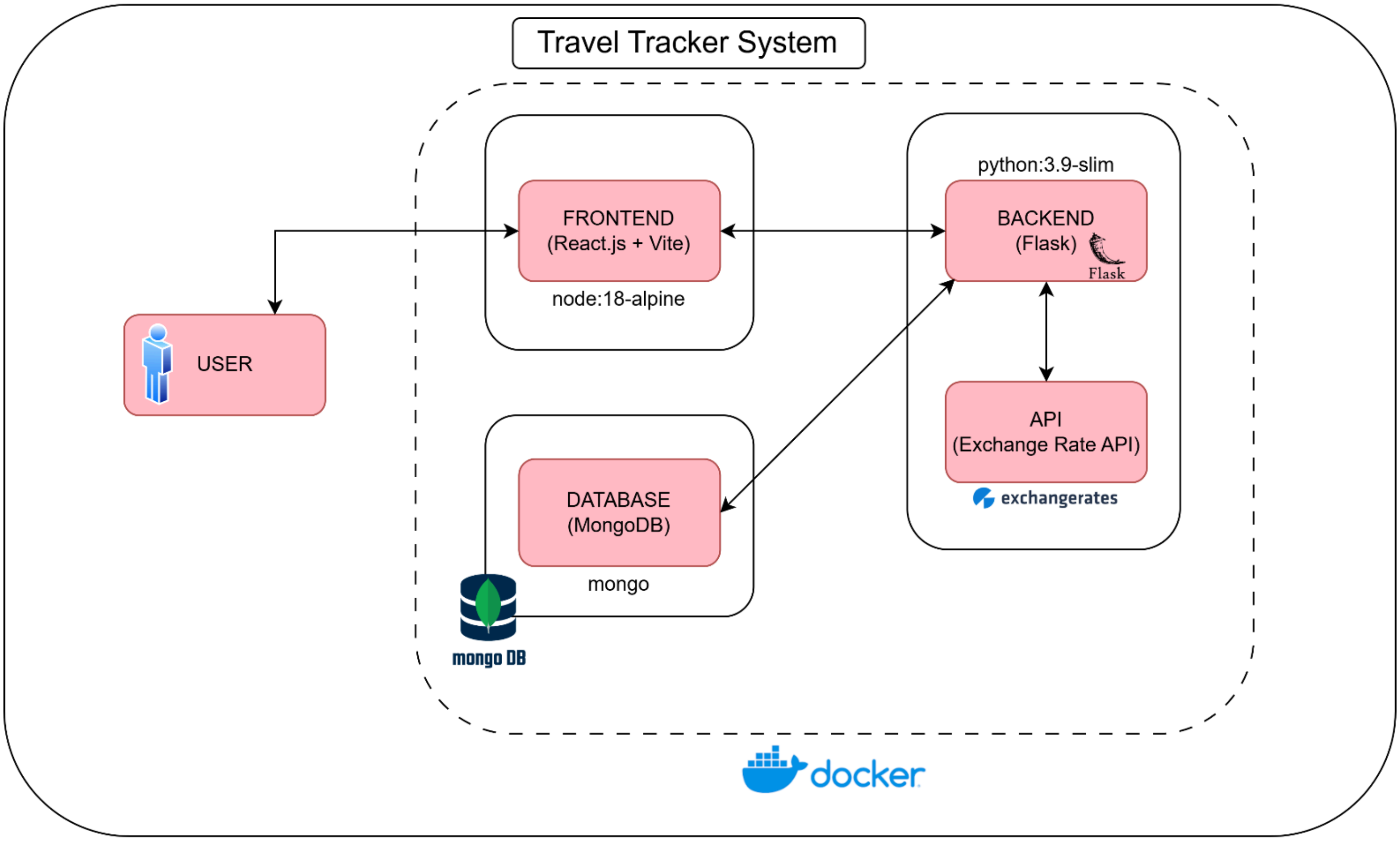
Planning vacations can be stressful and time-consuming, especially when managing multiple factors like expenses, itineraries, and currency conversions. Travelers often use separate tools to budget, convert currency, and organize their schedules, which can lead to inefficiencies and disorganization. Travel Tracker provides a user-friendly and secure platform that brings all these features together.

We plan to integrate expense tracking, budget management, itinerary planning, and real-time currency conversion into a single application :

1. Reduce travel planning stress through a centralized platform.
2. Enhance financial control by enabling users to track expenses, set allowances, and convert currencies seamlessly.
3. Improve organization with personalized itineraries and activity tracking.
4. Ensure security by safeguarding sensitive user data through robust authentication protocols.

This application encourages efficient planning, better financial management, and ultimately, a more enjoyable travel experience.

Architecture



Key Features:

- **User Authentication (Sign-up/Login)**
  - Users can create an account and log in securely using JWT authentication.
  - Ensures data privacy and secure access.
- **Vacation Budget Management**
  - Users can create a trip budget and track flight, hotel, food, and activity expenses.
  - The budgeting tool ensures expenses stay within limits
- **Expense & Banking Tracker**
  - Users can manually enter their income, expenses, and recurring payments. It helps in tracking finances efficiently.
- **Real-Time Currency Conversion**
  - Integrated with an *ExchangeRate API* to provide up-to-date currency conversions.
  - Users can convert expenses and budgets into different currencies.
- **Trip Itinerary & Activity Planner**
  - Users can add and organize their travel itineraries, including flights, hotel stays, and activities.
  - It helps users plan each day efficiently.
- **Notes & Additional Travel Info**
  - Users can add custom notes related to their trip.
  - A personalized section for important travel details.
- **Backend Data Storage & Management**
  - All user data, trip details, and transactions are securely stored in *MongoDB.*
  - Provides fast data retrieval and storage.
- **User Interface**
  - Elegant and user-friendly UI designed with *React.js*.
  - Simple navigation between budgeting, itinerary, and expenses.
- **Secure API Communication**
  - *Flask* handles all API requests, authentication, and interactions with MongoDB.
  - Ensures smooth data flow between the front end and the back end.

Chapter 2

Implementation

React: The client side of our application will be developed using **JavaScript** and **React**. This allows for functionality for user experience for easy logins, a clean user interface, and robust performance. Users can access and manage their trip details, such as budget, excursion plans, and notes. After logging in, users can input trip details stored in their user profiles. This data is synchronized with the Python-based backend through API calls. Changes in currency exchange rates will be up-to-date and displayed to the user through the ExchangeRate API. Users can monitor, edit, and save their trip plans and updates in real time.

Key Frontend Features:

- *Login/Signup Pages:* Secure authentication interfaces for user registration and login.
- *Tracker Page:* Users can log expenses, track budgets, and manage allowances. A drop-down menu allows users to convert finances into various currencies.
- *Travel Planner Page*: Users can create, manage, and edit vacation plans. The budgeting feature automatically compares expenses against the allocated budget and integrates with the currency conversion tool for international trips.
- *Currency Conversion Tool*: Real-time currency conversion using the ExchangeRate API for accurate budgeting and financial tracking.

MongoDB: MongoDB will handle user data. Each user profile will be stored, including username and password, and user inputs for budget, flight information, stay information, itinerary planning, personal notes, and interests.

Python: The backend will be developed using Python to manage server-side logic. Python will talk to the client about user data and communicate with the ExchangeRate API. This will use Flask. The back end will interact with MongoDB for data storage and retrieval.

API: To provide real-time currency conversion, we will integrate the **ExchangeRate API**. This API supplies up-to-date exchange rates for various global currencies, ensuring that users can accurately budget for international trips.

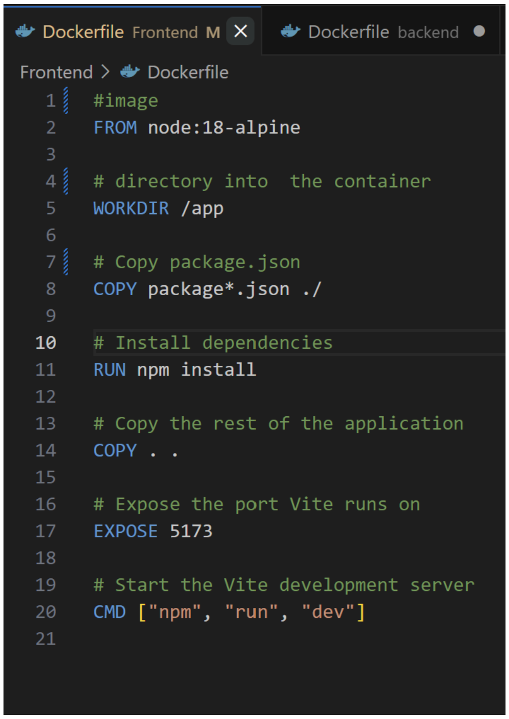
Key API Functions:

- *Currency Conversion*: Fetch current exchange rates based on user-selected currencies.
- *Automated Updates*: Regularly update exchange rates to reflect the latest market changes.
- *Integration with Expense Tracker*: Convert user-entered expenses into different currencies using API data.

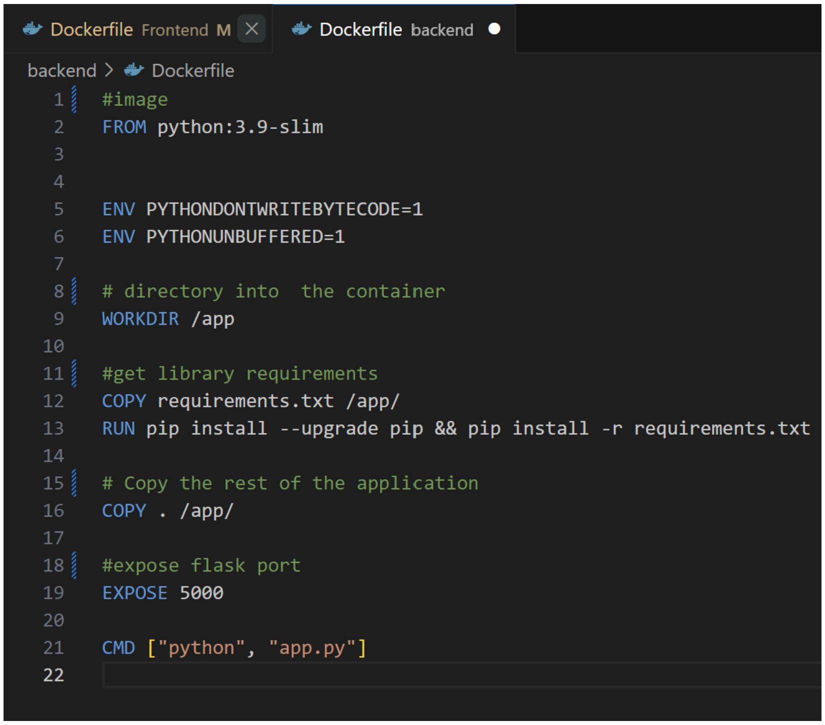
Chapter 3

Dockerization

Travel Tracker is containerized into three containers: frontend, backend, and MongoDB. Containers are reliant on the ability to talk to one another to ensure that the user experience and user interactions are sufficient. Our React frontend must be able to send HTTP requests to our Flask backend for users to be able to create their accounts, log in to their accounts, and create and modify their trip information. All user input must be able to be retained in our MongoDB database. In our dockerization process, we utilize Dockerfiles to build our front-end and back-end containers. Our front end uses the official image *node:18-alpine* as our base for a lightweight OS. Our backend uses the base image *python:3.9-slim* for a lightweight deployment, which works well for our Flask backend.



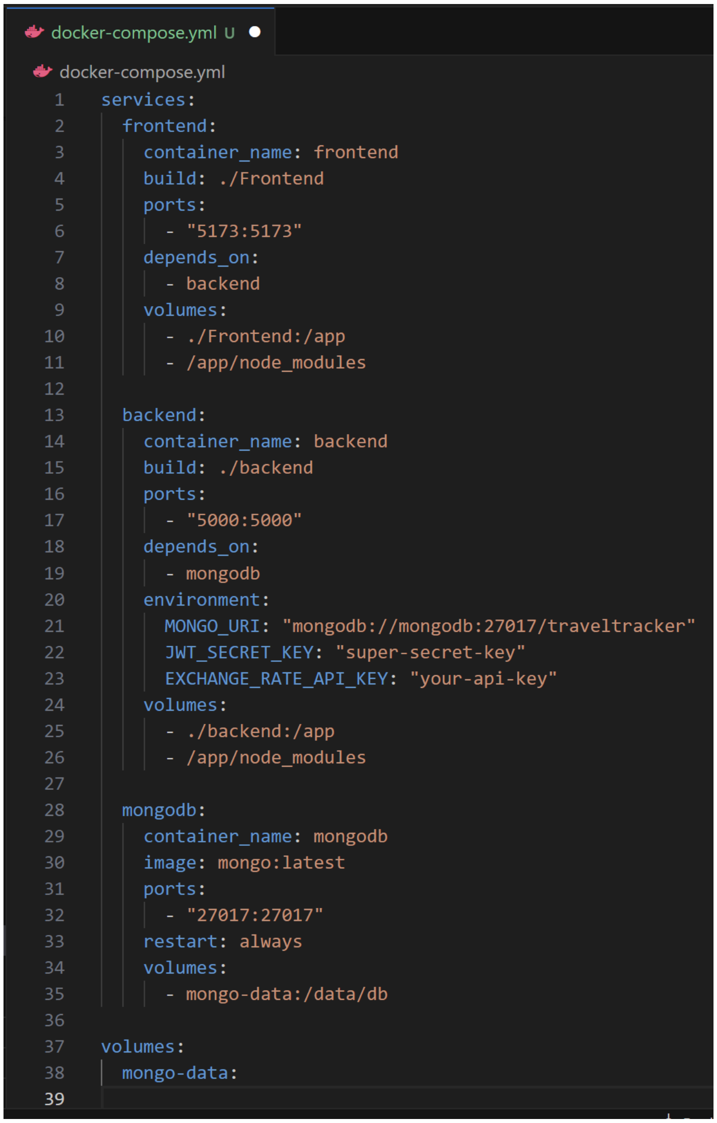
```dockerfile
#image
FROM node:18-alpine

# directory into  the container
WORKDIR /app

# Copy package.json
COPY package*.json ./

# Install dependencies
RUN npm install

# Copy the rest of the application
COPY . .

# Expose the port Vite runs on
EXPOSE 5173

# Start the Vite development server
CMD ["npm", "run", "dev"]
```



```dockerfile
#image
FROM python:3.9-slim


ENV PYTHONDONTWRITEBYTECODE=1
ENV PYTHONUNBUFFERED=1

# directory into  the container
WORKDIR /app

#get library requirements
COPY requirements.txt /app/
RUN pip install --upgrade pip && pip install -r requirements.txt

# Copy the rest of the application
COPY . /app/

#expose flask port
EXPOSE 5000

CMD ["python", "app.py"]
```

These Dockerfiles are built by our docker-compose.yml, which calls for both the ./frontend and ./backend builds. Docker-compose.yml is responsible for building our MongoDB database container. This container uses the official image for a MongoDB database, *mongo*.

Travel Tracker's data is primarily collected through user interactions. Users create accounts, log in, and create/edit trip information. Our schema is specified in the backend configuration using an init.js file. In the future, we intend to supply the database with a sample data python script that, upon the container's first initialization, our database will be ready for the backend interactions. Data collection occurs for all saved user input, like account information and trip information. Data is sent from the front end through a POST request to the backend and then stored in the collections in our MongoDB database.



```yaml
services:
  frontend:
    container_name: frontend
    build: ./Frontend
    ports:
      - "5173:5173"
    depends_on:
      - backend
    volumes:
      - ./Frontend:/app
      - /app/node_modules

  backend:
    container_name: backend
    build: ./backend
    ports:
      - "5000:5000"
    depends_on:
      - mongodb
    environment:
      MONGO_URI: "mongodb://mongodb:27017/traveltracker"
      JWT_SECRET_KEY: "super-secret-key"
      EXCHANGE_RATE_API_KEY: "your-api-key"
    volumes:
      - ./backend:/app
      - /app/node_modules

  mongodb:
    container_name: mongodb
    image: mongo:latest
    ports:
      - "27017:27017"
    restart: always
    volumes:
      - mongo-data:/data/db

volumes:
  mongo-data:
```

Our database is structured into two collections: *users* and *trips*. As MongoDB stores individual records as documents, our collections are organized by user ID. For each new trip a user adds, a new document is added to the trips collection. The *user's* collection specifies username, email, and password. The *trips* collection formats each trip so that the user can specify their budget, different expenses like flight and hotel, itinerary, and notes. It is important that users have the functionality to not be restricted to a specified number of itinerary entries or character length for notes. Data is stored persistently, and the use of Docker volumes ensures that data is not lost during container restarts.

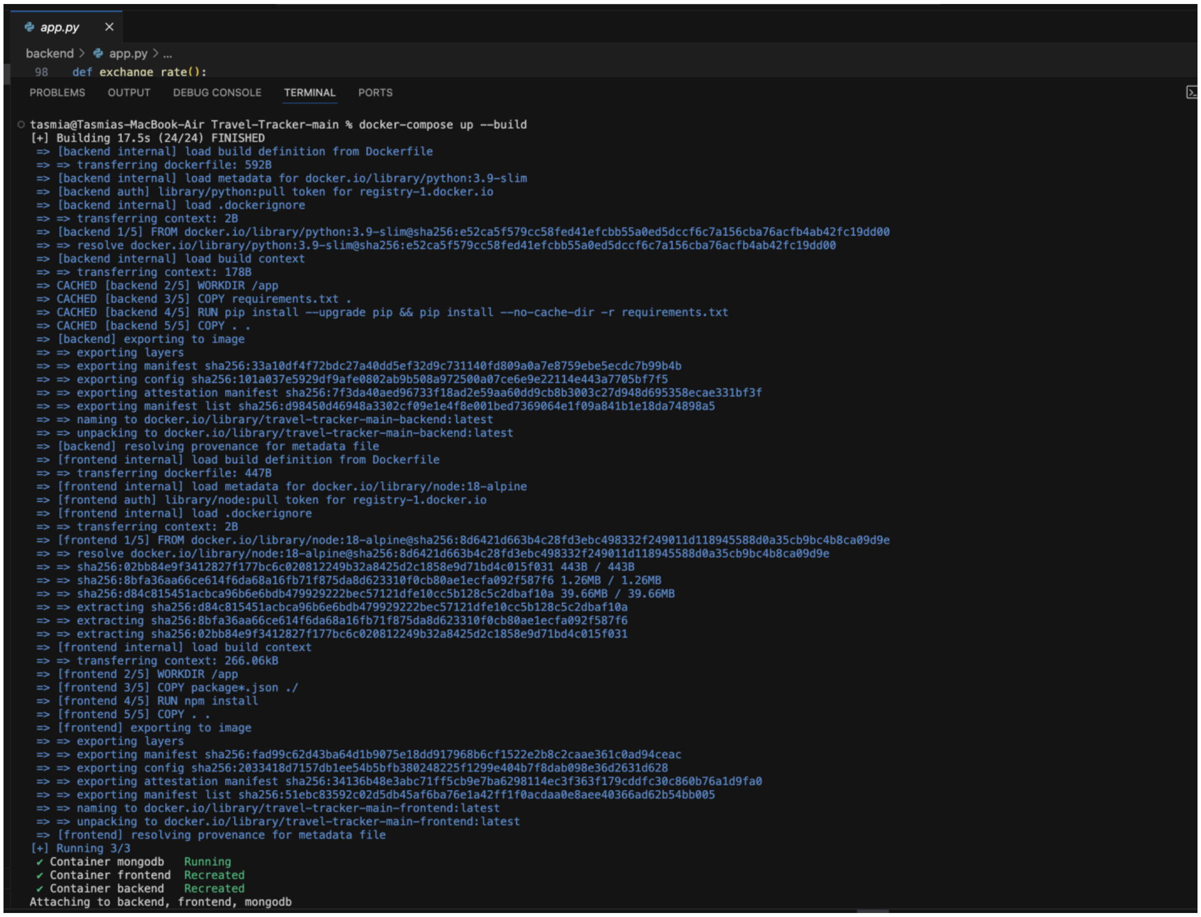
Our database must persist; thus, our volume configuration stores data in mongo-data. We also use volumes for our front-end and back-end containers. Specifying the volumes allows persistent data to ensure that user data is kept. For example, the */app/node_modules* volume helps keep dependencies intact. By specifying *depends_on*, we know that the backend must wait for the frontend and MongoDB to be ready before starting. This helps avoid race conditions when one service is waiting for another to be up.

To test container communication, we simulated user login requests to ensure that the frontend can make requests, the backend can process the requests, and MongoDB responds. In addition to this, we established a simple test button to quickly establish that the front-end and back-end could talk to each other. Preliminary results show us that our user data is being stored.
Our team has successfully built our images and has found success in running the services. The simplicity of *Docker Compose build* and *Docker Compose up* allows for the efficient deployment of Travel Tracker.

Services: Defines the individual services (frontend, backend, MongoDB).

- **frontend**: Runs the frontend application, exposes port 5173, and depends on the backend.
- **backend**: Runs the backend API, exposes port 5000, depends on MongoDB, and uses environment variables to configure MongoDB and API keys.
- **MongoDB**: Runs the MongoDB database, stores data persistently using a mounted volume, and exposes port 27017.

Our Docker setup provides an efficient way to containerize and manage a multi-service application, including the front-end, back-end, and database. Docker Compose simplifies running all these services together with clear configuration, and the use of Dockerfiles ensures consistency across development and production environments. We are currently using Docker Compose for local development and multi-container orchestration. While Kubernetes is our intended deployment tool, we are still finalizing component integration and data validation. Full deployment and Kubernetes YAMLs will be prepared in the next stage.



```
def exchange_rate():
```

```
○ tasmia@Tasmias-MacBook-Air Travel-Tracker-main % docker-compose up --build
  [+] Building 17.5s (24/24) FINISHED
   => [backend internal] load build definition from Dockerfile
   => => transferring dockerfile: 592B
   => [backend internal] load metadata for docker.io/library/python:3.9-slim
   => [backend auth] library/python:pull token for registry-1.docker.io
   => [backend internal] load .dockerignore
   => => transferring context: 2B
   => [backend 1/5] FROM docker.io/library/python:3.9-slim@sha256:e52ca5f579cc58fed41efcbb55a0ed5dccf6c7a156cba76acfb4ab42fc19dd00
   => => resolve docker.io/library/python:3.9-slim@sha256:e52ca5f579cc58fed41efcbb55a0ed5dccf6c7a156cba76acfb4ab42fc19dd00
   => [backend internal] load build context
   => => transferring context: 178B
   => CACHED [backend 2/5] WORKDIR /app
   => CACHED [backend 3/5] COPY requirements.txt .
   => CACHED [backend 4/5] RUN pip install --upgrade pip && pip install --no-cache-dir -r requirements.txt
   => CACHED [backend 5/5] COPY . .
   => [backend] exporting to image
   => => exporting layers
   => => exporting manifest sha256:33a10df4f72bdc27a40dd5ef32d9c731140fd809a0a7e8759ebe5ecdc7b99b4b
   => => exporting config sha256:101a037e5929df9afe0802ab9b508a972500a07ce6e9e22114e443a7705bf7f5
   => => exporting attestation manifest sha256:7f3da40aed96733f18ad2e59aa60dd9cb8b3003c27d948d695358ecae331bf3f
   => => exporting manifest list sha256:d98450d46948a3302cf09e1e4f8e001bed7369064e1f09a841b1e18da74898a5
   => => naming to docker.io/library/travel-tracker-main-backend:latest
   => => unpacking to docker.io/library/travel-tracker-main-backend:latest
   => [backend] resolving provenance for metadata file
   => [frontend internal] load build definition from Dockerfile
   => => transferring dockerfile: 447B
   => [frontend internal] load metadata for docker.io/library/node:18-alpine
   => [frontend auth] library/node:pull token for registry-1.docker.io
   => [frontend internal] load .dockerignore
   => => transferring context: 2B
   => [frontend 1/5] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e
   => => resolve docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e
   => => sha256:02bb84e9f3412827f177bc6c020812249b32a8425d2c1858e9d71bd4c015f031 443B / 443B
   => => sha256:8bfa36aa66ce614f6da68a16fb71f875da8d623310f0cb80ae1ecfa092f587f6 1.26MB / 1.26MB
   => => sha256:d84c815451acbca96b6e6bdb479929222bec57121dfe10cc5b128c5c2dbaf10a 39.66MB / 39.66MB
   => => extracting sha256:d84c815451acbca96b6e6bdb479929222bec57121dfe10cc5b128c5c2dbaf10a
   => => extracting sha256:8bfa36aa66ce614f6da68a16fb71f875da8d623310f0cb80ae1ecfa092f587f6
   => => extracting sha256:02bb84e9f3412827f177bc6c020812249b32a8425d2c1858e9d71bd4c015f031
   => [frontend internal] load build context
   => => transferring context: 266.06kB
   => [frontend 2/5] WORKDIR /app
   => [frontend 3/5] COPY package*.json ./
   => [frontend 4/5] RUN npm install
   => [frontend 5/5] COPY . .
   => [frontend] exporting to image
   => => exporting layers
   => => exporting manifest sha256:fad99c62d43ba64d1b9075e18dd917968b6cf1522e2b8c2caae361c0ad94ceac
   => => exporting config sha256:2033418d7157db1ee54b5bfb380248225f1299e404b7f8dab098e36d2631d628
   => => exporting attestation manifest sha256:34136b48e3abc71ff5cb9e7ba6298114ec3f363f179cddfc30c860b76a1d9fa0
   => => exporting manifest list sha256:51ebc83592c02d5db45af6ba76e1a42ff1f0acdaa0e8aee40366ad62b54bb005
   => => naming to docker.io/library/travel-tracker-main-frontend:latest
   => => unpacking to docker.io/library/travel-tracker-main-frontend:latest
   => [frontend] resolving provenance for metadata file
  [+] Running 3/3
   ✔ Container mongodb   Running
   ✔ Container frontend  Recreated
   ✔ Container backend   Recreated
  Attaching to backend, frontend, mongodb
```

Chapter 4: Final Results

By the end of the project timeline, the Travel Tracker application successfully achieved the majority of its intended goals. The final system consists of three containerized services: frontend (React), backend (Flask/Python), and a MongoDB database, managed through Docker Compose. The frontend allows users to sign up, log in, create and manage travel plans, track expenses, and convert currencies in real-time. The backend ensures secure authentication, handles all API interactions, and manages the application's business logic. MongoDB provides persistent storage for user and trip data, maintaining security and organization of critical information.

Key Functionalities Implemented

- **Containerized Deployment:** Frontend, backend, and database running in isolated, scalable containers through Docker Compose.
- **User Authentication:** Full signup and login capabilities with JWT authentication.
- **Vacation Budget Management:** Ability to set travel budgets, record expenses (flight, hotel, food, activities), and monitor budget status.
- **Expense Tracker:** Manual entry of expenses and income with support for recurring transactions.
- **Currency Conversion:** Real-time currency updates using the ExchangeRate API, integrated into both expense tracking and budgeting modules.
- **Persistent Database Storage:** MongoDB volumes are configured to maintain data across container restarts.

Challenges Faced

- **Updated Frontend With Schema Changes:** After improving our frontend with a more complex design, we needed to change how we intended to store user and trip data to connect the backend to the frontend and ensure data was being saved. Our code had difficulty adapting to the change, and we needed to work around the problem.
- **Time Constraint:** This semester was particularly challenging so we often had a hard time finding meetup times within this time constraint.

Missed Milestones

- **No AI Implementation**
- **Late Final Completion in the Overall Timeline**
- **Kubernetes Implementation**

Testing and Validation

Testing was performed across all major functionalities to ensure proper communication between the frontend, backend, and database components. Key testing activities included:

- Simulated user account creation, login, and session management.
- CRUD operations for trips and expenses.
- Currency conversion accuracy checks using live API data.
- Container resilience testing by restarting services and validating persistent data.

While full-scale automated testing was not completed, manual testing was thorough across multiple use cases.

Future Ideas

In the future, the following ideas will propel Travel Tracker's complexity, intrigue, and industry-competitiveness:

- **Kubernetes Deployment:** Completing the implementation of Kubernetes would make the application even more scalable and production-ready.
- **Improved UI/UX:** A polished design enables the user to be excited by the application, especially to make it mobile-friendly.
- **Automated Testing:** Building a full set of unit and integration tests to ensure smooth deployment, testing, and quality testing.
- **Trip Sharing/Other Tabs:** Letting users share their trips with friends or plan group vacations.
- **AI Recommendations:** Integrating a large language model to suggest hotels, activities, or destinations based on a user's budget and preferences would improve user experience and alleviate trip planning stress.
- **Advanced Uploads:** Allowing users to upload images or tickets for flights would enable advanced trip planning. Creating a hub for all the user's information with more advanced fields for flights, stays, and itinerary would benefit the user.

Conclusion

Creating, designing, and building Travel Tracker has been a rewarding learning experience for our team. Over the course of the project, we built a full-stack web application from the ground up, in which the team learned the hands-on skills of cloud deployment and web development and also the importance of teamwork and communication when working toward a goal.

One of the biggest takeaways from this project is how powerful containerization can be. Using Docker and Docker Compose simplified how we manage our different services: frontend, backend, and database in a consistent way that is replicable and scalable. We also learned about how APIs work in real-time applications, how to handle user authentication securely, and how to store and manage user data properly with MongoDB. There were some challenges along the way, like reformatting our project's endpoints and managing the different dependencies required for the containers when they spin up. Working through these issues helped us improve our problem-solving, and we adapted in the event a challenge arose.

Overall, we are proud of the current state of Travel Tracker and our work to create it. Travel Tracker is a solid foundation that has potential to be expanded upon. Most importantly with this project, we learned through a real-world application of building an application to be deployed on the cloud. We learned through not just writing code, but designing systems, solving unexpected problems, and working together as a team.

Olivia Folsom

Folsom019@gmail.com | www.linkedin.com/in/olivia-folsom | Chester Springs, PA

Bringing a unique blend of technical expertise and strong interpersonal skills with a dual major in Computer Science and German, along with a passion for continuous learning. Studies in computer science have sharpened problem-solving abilities while a background in German has enhanced communication and analytical skills. Seeking opportunities in software engineering and front-end development to contribute to creating innovative, user-centric solutions.

Education

WEST CHESTER UNIVERSITY OF PENNSYLVANIA | West Chester, PA | Jan 2022 – Expected Dec 2025

Bachelor of Science in Computer Science and Bachelor of Arts in German Language

- Dean's List Recipient, all semesters
- Relevant Coursework: Software Engineering, Cloud Computing, Modern Web Applications Using Server-Side Technologies, Artificial Intelligence, Data Structures and Algorithms, Discrete Mathematics

ACTILINGUA ACADEMY | Vienna, Austria | Jun 2023 – Jul 2023

Study Abroad Program

- Completed intensive German language courses at B1 and B2 levels, improving fluency, comprehension, and communication skills in both written and spoken German.
- Gained deep insights into Austrian history and dialects, actively participating in cultural activities and lifestyle with native speakers.

Professional Experience

WEST CHESTER UNIVERSITY OF PENNSYLVANIA | West Chester, PA | Aug 2024 – Present

Computer Science Tutor

- Provide academic support to students in undergraduate computer science courses from introductory courses to computer systems and data structures.
- Monitor student progress, provide constructive feedback, and collaborate with faculty to meet students' needs. Teach students to debug code and manage assignments.

INCYTE CORPORATION | Wilmington, DE | Jun 2024 – Jul 2024

IT Security Intern

- Initiated research of AI models and generative AI (GenAI) applications, with a focus on responsible AI practices and their integration into software solutions and company productivity.
- Performed an in-depth risk assessment of Incyte's GenAI initiatives, evaluating potential security concerns from multiple angles, meeting with key figures and users in the company, including end-users of GenAI applications and developers utilizing the ChatGPT API and ChatGPT Enterprise. Established a plan for documentation and awareness training required for company use of GenAI applications.
- Spearheaded the creation of an in-house document on responsible AI usage for end-users, establishing guidelines for secure GenAI deployment and usage.
- Conducted and led a high-impact live risk awareness training session for 240 company-wide participants, significantly advancing the organization's understanding of AI-related risks.

TARGET | Exton, PA | Oct 2021 – Jun 2024

Guest Service Advocate

- Provided positive, efficient customer service, processed sales transactions and merchandise returns, resolved issues with merchandise, and managed and operated Target's Drive-Up service.
- Trained and guided new team members in Target's point-of-sale technology and best practices for customer interactions.

Extracurricular Experience

WEST CHESTER UNIVERSITY LANGUAGE DEPARTMENT | West Chester, PA | May 2023 – Present

Ambassador

- Assisted at the global Austrian Studies Association Conference 2024 by welcoming visiting scholars, providing information on event schedules and locations, distributing event materials, and ensuring smooth event operations.
- Engaged with community and students through annual language fairs. Informed prospective language students about the German language program, study abroad programs, and taught cultural and language facts.

WEST CHESTER UNIVERSITY PROGRAMMING CONTEST | West Chester, PA | Sep 2023 – Apr 2024

Competitor

- Participated in three programming contests, facing a range of coding challenges with diverse problem sets and developed and refined skills in algorithmic problem-solving and code optimization through competitive experiences.

WEST CHESTER UNIVERSITY ROBOTICS CAMP | West Chester, PA | Nov 2023

Participant

- Worked in a team to solve challenges with LEGO® MINDSTORMS® robotics, leveraging Python and block-coding, gaining hands-on experience with robotics and enhancing problem-solving skills.

Projects

Multiplayer Competitive Racing Game, Nov 2024

- Collaborated with a team to build a strategy game using Godot Game Engine with Object-oriented programming that supports 4-8 players where players race against one another, interacting with the map and other players in real time. Players compete for the best time to be displayed in a live leaderboard.
- Developed core multiplayer functionality, including user login and synchronization, generated map and map obstacle mechanics, and handled user data such as storing leaderboard and completion times.

Business Web Application, Jan 2025

- Developed a fully functional MVC-compliant web application, enabling users to input information and interact with business-related data.
- Employed a full-stack development approach, utilizing JavaScript for the front end, Node.js for server-side operations, and MongoDB for efficient storage and retrieval of user input.

Certifications

AWS Certified Cloud Practitioner, Amazon Web Services (AWS), Jul 2024

- Acquired foundational knowledge in cloud computing and core AWS services and developed a comprehen understanding of IT technologies and infrastructure, enhancing ability to support cloud-based solutions ar strategies.

Skills

- **Natural Languages:** English, German
- **Programming Languages:** Java, Python, Java, C, JavaScript, Haskell
- **Web Development:** React, Node.js, Express.js, HTML/CSS, RESTful APIs
- **Software Development:** Git, Docker, Visual Studio Code, Agile
- **Game Development:** Godot Game Engine with GDScript, Unreal Engine, Blender
- **Cloud Computing:** AWS
- **Design Tools**: Adobe Photoshop, Illustrator, After Effects

Tasmia Iqbal

2157306739 | tasmia203@gmail.com | Upper Darby, PA

PROFILE

Resourceful and dedicated Computer Science major with a solid background in software development, technical support, and engineering fundamentals. Proficient in multiple programming languages including Java, Python, C/C++, and JavaScript, with experience in web development using HTML, Node.js, and Express.js. Skilled in problem-solving, debugging, and delivering technical solutions in both academic and professional settings. Strong communicator with a commitment to continuous learning and developing innovative applications.

EDUCATION

Bachelor of Science in Computer Science

West Chester University | West Chester, PA

May 2025

Associate of Science Degree in Engineering

Delaware County Community College | Media, PA

August 2020- May 2022

Coursework: Calculus I, II, III; University Physics I, II, Engineering Graphics, Engineering Topics, Introduction to C++, Engineering Statics, Differential Equations

EXPERIENCE

IS&T Help Desk

January 2024

West Chester University | West Chester, PA

- Help student with troubleshoot their PC or MacBook
- Front end worker who helps student to find a technical solution on Customer Laptop
- Help students resolve Microsoft, Excel, Adobe, and recording issues on their laptop or computer.

Customer Service Center

September 2021

Delaware County Community College | Media, PA

- Communicated with students to help them with needs and desires
- Guided students and encourage onward education
- Performed problem solving skills to provide information to students

Team Member

June 2019-2021

Wendy's | Upper Darby, PA

- Communicated with customers to meet their needs
- Helped bag orders in a timely manner
- Collaborated with coworkers on how to functionally run store

SKILLS AND ABILITIES

- Problem Solving & Troubleshooting: Proficient at identifying and resolving complex technical issues in hardware and software environments.
- Software Development: Experienced in developing and maintaining applications using Java, Python, and JavaScript.
- Communication & Collaboration: Strong interpersonal skills with experience in technical support and teamwork across various roles.
- Adaptability: Quickly learns new technologies and adapts to fast-paced environments.
- Customer Service: Excellent ability to communicate technical concepts clearly to non-technical users, ensuring effective support.

TECHNICAL SKILLS

- Languages: Java, Python, C, C++, JavaScript, Node.js, HTML, CSS
- Frameworks & Libraries: Express.js
- Web Development: RESTful APIs, JSON, Axios
- Database Management: MongoDB, MySQL (Basic Knowledge)
- Tools & Platforms: GitHub, VS Code, Jupyter Notebook, IntelliJ
- Concepts: Object-Oriented Programming (OOP), Frontend & Backend Development, UI/UX Design Principles, API Integration

COMMUNITY SERVICE / EXTRACURRICULAR INVOLVED

- College Possible *(2016 – Present)*: Mentored peers in academic growth and provided leadership support for educational initiatives.
- American Red Cross Volunteer *(2017 – Present)*: Assisted in community outreach and organized local fundraising events.
- Computer Science Club *(2023 – Present)*: Participated in coding workshops, hackathons, and collaborative tech projects.
- CrossFit Club *(2024 – Present)*: Promoted teamwork, discipline, and health through fitness activities.
- RECAP Conference Attendee *(September 2024)*: Attended sessions on software development, cloud computing, and emerging tech trends.

PANHAPICH LEANG

COMPUTER SCIENCE STUDENT

571.585.2286 | PHILADELPHIA, PA | LEANGPANHAPICH2710@YAHOO.COM|
HTTPS://WWW.LINKEDIN.COM/IN/PANHAPICH-LEANG/
HTTPS://GITHUB.COM/PANHAPICHLEANG

Profile

Motivated Computer Science student at West Chester University with dual Associate Degrees in Business and Computer Information Systems. Proficient in programming algorithms, Object-Oriented Programming (OOP), and software development practices. Demonstrates strong problem-solving skills, a dedicated work ethic, and effective time management. Passionate about pursuing a career in Software Engineering with a focus on code development and system optimization.

Skills & abilities

- **Programming Languages:** Java, Python
- **Familiarity With:** C, JavaScript, SQL, CSS, HTML
- **Tools & Platforms:** MS Word, Excel, PowerPoint, GitHub, Visual Studio Code
- **Soft Skills:** Time Management, Problem Solving, Communication
- **Other Skills:** Bilingual, Keyboard Typing Proficiency

Experience

TEXT PROCESSING PROJECT | CSC240| WEST CHESTER UNIVERSITY, PA|FALL 2024

- **Problem Solving:** Analyzed project requirements and developed a structured approach to create a text processing application in Java.
- **Technical Implementation:** Implemented Java file I/O operations to read, manipulate, and write text data effectively.
- **Design & Planning:** Created a comprehensive UML diagram to visualize program structure and workflows.
- **Documentation:** Produced detailed reports outlining the development process, challenges faced, and solutions applied.
- **Outcome:** Delivered a fully functional program that met all requirements, showcasing strong programming and analytical skills.

LINKEDLIST EXPLORATION PROJECT | CSC240 | WEST CHESTER UNIVERSITY, PA | FALL 2024

- **Innovative Development:** Designed and implemented a custom doubly linked list in Java with advanced features for performance optimization.
- **User Experience:** Developed an interactive GUI to make the data structure visualization user-friendly and engaging.
- **Planning & Testing:** Constructed UML diagrams to map the system architecture and executed rigorous testing to validate functionality.
- **Analytical Insights:** Documented testing results, reflected on program behavior, and provided a narrative of the development process.
- **Achievement:** Delivered an efficient, interactive program that enhanced understanding of data structures and GUI implementation.

Education

WEST CHESTER UNIVERSITY | WEST CHESTER , PA

January 2024 – present

Pursuing Bachelor Degree in Computer Science
Grade Level: Junior
Expected Graduation: May 2026

COMMUNITY COLLEGE OF PHILADELPHIA | PHILADELPHIA , PA

September 2019 – December 2023

Associate Degree in Business General
Associate Degree in Computer Information System

OLIVIA TARSILLO

Software Engineer

DETAILS

ADDRESS

West Chester
United States

PHONE

848-448-2249

EMAIL

otarsillo@gmail.com

LINKS

LinkedIn : Olivia Tarsillo

SKILLS

Java

Adobe Photoshop

Communication Skills

Ability to Work in a Team

Python

Microsoft Excel

Robotics

Haskell

C

HTML & CSS

PROFILE

An ambitious and dedicated computer programmer with years of experience in activism in STEM. Skilled in Java with a strong foundation in Python, C, Haskell, and Adobe Photoshop. Adept at working in a dynamic team environment and communicating effectively to ensure the best customer experience. Eager to leverage my skills and passion for system design and gaming into a software engineering internship, while continuing to expand my programming expertise. Aiming to contribute positively to a forward-thinking organization in a challenging and rewarding role.

EDUCATION

Bachelors, West Chester University ~ GPA: 3.6

West Chester

2022 — Present

Relevant Coursework: Computer Science I, Computer Science II, Computer Science III, Computer Systems, Data Structures and Algorithms, Foundations of Computer Science, Computer Security and Ethics, Programming Language Concepts and Paradigms

EXTRA-CURRICULAR ACTIVITIES

Women in Computer Science Club, Vice President At West Chester University West Chester, PA

West Chester

Sep 2022 — Present

- Participate in group functions promoting women in computer science.
- Educated high school girls on basic software engineering methods.
- Work on group projects promoting computer science career paths.
- Hold group meetings as both an opportunity and a safe space for women on campus.
- Help lead and create a successful organization.

Robotic Competition, Host of Robotics Camp At West Chester University West Chester, PA

West Chester

Nov 2023 — Nov 2024

- Assisted and taught fourth-year computer science majors on functions of robotics.
- Used Python and scratch-level coding for proficient and starting programmers.

- Hosted the event in a group to make intelligent courses for the robots to test.
- Both a competition and a camp for majors and non-majors to learn coding and robotics.

Super Science Saturday , Teacher and Volunteer

West Chester

Oct 2024 — Present

- Taught girls aged 10-16 about the importance of women in STEM and Computer Science.
- Created simple scratch code competitions for young women to utilize Lego robots.
- Influenced young women to branch out into the opportunities of programming.

EMPLOYMENT HISTORY

West Chester Recreational Center - Rec Attendant, Dan Comas

West Chester

Aug 2024 — Present

- Organized and supervised recreational activities to ensure clients' safety and maximize their enjoyment.
- Greeted customers in a friendly and professional manner
- Assisted with locating items and providing product information
- Handled customer complaints calmly and professionally, resolving customer issues promptly

Forever 21 - Sales Associate

East Brunswick

Feb 2021 — Sep 2022

- Executing all retail activities related to merchandise sales and customer engagement.
- Maintained store selling floor, fitting rooms, marketing displays, and conditions supporting up-sell opportunities.
- Inventory and merchandise management in support of directional sales goals.
- Engaged with a dynamic team of sales employees and management professionals focused on productivity.
- Excellent communication skills.

PROJECTS

Kitty Kat Adventure

2023 — Present

https://github.com/LivTheKittyLover/Kitty-Kat-Adventure

Spam or Ham Word Email Learning Tabulator

2023

https://github.com/LivTheKittyLover/Spam-or-Ham